



# Localisation et cartographie simultanées en environnement extérieur à partir de données issues d'un radar panoramique hyperfréquence

Franck Gérossier

## ► To cite this version:

Franck Gérossier. Localisation et cartographie simultanées en environnement extérieur à partir de données issues d'un radar panoramique hyperfréquence. Autre. Université Blaise Pascal - Clermont-Ferrand II, 2012. Français. NNT : 2012CLF22223 . tel-00864181

**HAL Id: tel-00864181**

**<https://theses.hal.science/tel-00864181>**

Submitted on 20 Sep 2013

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N° d'ordre : D.U : 2243  
EDSPIC : 561

UNIVERSITÉ BLAISE PASCAL - CLERMONT II

ÉCOLE DOCTORALE  
SCIENCES POUR L'INGÉNIEUR DE CLERMONT-FERRAND

# Thèse

Présentée par  
**FRANCK GÉROSSIER**

Pour obtenir le grade de  
**DOCTEUR D'UNIVERSITÉ**  
SPÉCIALITÉ : Vision pour la Robotique

**Localisation et cartographie simultanées en  
environnement extérieur à partir de données issues  
d'un radar panoramique hyperfréquence**

Soutenue publiquement le 5 juin 2012 devant le jury :

<i>Président :</i>	Dominique MEIZEL	-	Professeur ENSIL/UL
<i>Rapporteurs :</i>	François GOULETTE	-	Professeur ENS MINES ParisTech
	Simon LACROIX	-	Directeur de recherche CNRS/LAAS
<i>Examineurs :</i>	Roland CHAPUIS	-	Professeur
			Université Blaise Pascal/Institut Pascal
	Paul CHECCHIN	-	Maître de conférences
			Université Blaise Pascal/Institut Pascal
<i>Invités :</i>	Philippe GOSSET	-	Chargé de projet THALES GROUP
	Patrice FAURE	-	Ingénieur d'étude Irstea
<i>Directeur :</i>	Laurent TRASSOUDAIN	-	Professeur
			Université Blaise Pascal/Institut Pascal

---

# Remerciements

---

En préambule à ce mémoire, je souhaite adresser mes remerciements les plus sincères aux personnes qui m'ont apporté leur aide et qui ont contribué à l'élaboration de ce travail.

Je tiens à remercier plus particulièrement Laurent Trassoudaine qui, en tant que directeur de thèse, s'est toujours montré à l'écoute. Ses conseils et le temps qu'il a bien voulu me consacrer m'ont permis de travailler dans les meilleures conditions possibles.

Mes remerciements s'adressent également à Paul Checchin pour son implication dans les orientations de mes travaux et pour son soutien sans faille.

Je remercie aussi Roland Chapuis pour ses conseils avisés ainsi que toutes les personnes de la société THALES, d'IRSTEA, et de l'institut Pascal, qui ont collaboré à la réalisation de cette tâche.

Je remercie également les rapporteurs de cette thèse : François Goulette et Simon Lacroix pour l'intérêt qu'ils ont porté à mon travail. Merci également aux autres membres du jury qui ont accepté de juger ce travail : Dominique Meizel, Philippe Gosset et Patrice Faure.

Enfin, j'adresse mes remerciements à Marie-Odile, Damien, Patrice, Raphael, Maxime, Philippe ... qui ont grandement participé à la réussite de cette thèse.

Je n'oublie pas ma famille. Elliott, Juliette, Marie-Élise et Emmanuelle, pour l'amour qu'ils portent à quelqu'un qui travaille souvent tard le soir.

Mille excuses à ma femme Emmanuelle pour avoir confondu notre domicile avec mon bureau.

Merci à toutes et à tous.

---



# Résumé

---

Le SLAM, « *Simultaneous Localisation And Mapping* », représente à l'heure actuelle l'une des principales thématiques investiguées dans le domaine des robots mobiles autonomes. Il permet, à l'aide de capteurs extéroceptifs (laser, caméra, radar, etc.) et proprioceptifs (odomètre, gyromètre, etc.), de trouver l'orientation et la localisation d'un robot dans un environnement extérieur vaste, inconnu ou modifié, avec la possibilité de créer une carte au fur et à mesure des déplacements du véhicule.

Les travaux de thèse décrits dans ce manuscrit s'intègrent dans ce courant de recherche. Ils visent à développer un SLAM innovant qui utilise un radar à modulation de fréquence continue « *FMCW* » comme capteur extéroceptif.

Ce capteur est insensible aux conditions climatiques et possède une portée de détection importante. Néanmoins, c'est un capteur tournant qui, dans une utilisation mobile, va fournir des données corrompues par le déplacement du véhicule.

Pour mener à bien ces travaux, nous avons proposé différentes contributions :

- une correction de la distorsion par l'utilisation de capteurs extéroceptifs ;
  - le développement d'une technique de localisation et cartographie simultanées nommée RS-SLAM-FMT qui effectue un *scan matching* sur les observations et utilise un algorithme estimatif de type EKF-SLAM ;
  - l'utilisation, pour la première fois en SLAM, de la mise en correspondance par *Transformée de Fourier-Mellin* pour réaliser l'opération de *scan matching* ;
  - la création d'un outil expérimental pour déterminer la matrice de covariance associée aux observations ;
  - des tests de robustesse de l'algorithme dans des conditions d'utilisation réelles : dans des zones avec un faible nombre de points d'intérêts, sur des parcours effectués à vitesse élevée, dans des environnements péri-urbains avec une forte densité d'objets mobiles ;
  - la réalisation d'une application temps réel pour le test du procédé sur un véhicule d'exploration qui se déplace dans un environnement extérieur vaste.
-



# Table des matières

Remerciements . . . . .	iii
Résumé . . . . .	v
Table des matières . . . . .	vii
<b>Liste des Figures</b>	<b>xi</b>
Table des figures . . . . .	xi
<b>Liste des Tables</b>	<b>xv</b>
Liste des tableaux . . . . .	xv
<b>1 Introduction</b>	<b>1</b>
1 Bref historique des outils de navigation . . . . .	2
2 Les outils de la navigation moderne . . . . .	6
3 Limites du positionnement global . . . . .	7
4 De l'homme au robot autonome . . . . .	7
5 Localisation autonome et cartographie . . . . .	8
5.1 Capteurs . . . . .	8
5.1.1 Les capteurs proprioceptifs . . . . .	9
5.1.2 Les capteurs extéroceptifs . . . . .	9
5.2 Localisation . . . . .	11
5.2.1 Localisation à l'estime . . . . .	11
5.2.2 Localisation absolue avec une carte <i>a priori</i> . . . . .	11
5.2.3 Localisation absolue sans carte <i>a priori</i> . . . . .	13
5.3 Cartographie . . . . .	14
6 Contributions scientifiques . . . . .	15
7 Organisation du manuscrit . . . . .	16
<b>2 Les données du radar K2Pi</b>	<b>19</b>
1 Introduction . . . . .	19
2 Principe du radar <i>FMCW</i> . . . . .	19
2.1 Effet Doppler . . . . .	21
2.2 Le speckle . . . . .	21
3 Caractéristiques du radar K2Pi . . . . .	22
4 Calcul d'un panoramique corrigé . . . . .	23
4.1 Utilisation des capteurs tournants dans la robotique mobile . . . . .	23
4.2 Construction d'un panoramique corrigé . . . . .	25
4.3 Correction de l'effet Doppler . . . . .	27
4.4 Correction du cône de détection . . . . .	28
4.5 Correction du speckle . . . . .	30



4.6	Conclusion . . . . .	30
<b>3</b>	<b>SLAM - État de l'art</b>	<b>33</b>
1	Introduction . . . . .	33
2	Bref historique du SLAM . . . . .	34
3	Orientation de l'étude bibliographique . . . . .	36
3.1	L'EKF-SLAM comme algorithme estimatif . . . . .	36
3.2	Le capteur extéroceptif dans la littérature SLAM . . . . .	37
3.3	Taxinomie retenue . . . . .	37
4	Etat de l'art . . . . .	38
4.1	Introduction . . . . .	38
4.2	SLAM avec une représentation par grille d'occupation . . . . .	38
4.2.1	Conclusion . . . . .	40
4.3	SLAM fondé sur une représentation topologique . . . . .	40
4.3.1	Conclusion . . . . .	42
4.4	SLAM avec une carte d'amers . . . . .	42
4.4.1	Les capteurs extéroceptifs . . . . .	43
4.4.2	Algorithmes d'estimation . . . . .	45
4.4.3	Conclusion . . . . .	46
4.5	SLAM orienté trajectoire . . . . .	46
4.5.1	Méthodes d'estimation . . . . .	47
4.5.2	<i>Scan matching</i> . . . . .	49
4.5.3	Conclusion . . . . .	56
5	Conclusion . . . . .	57
<b>4</b>	<b>Cartographie et localisation simultanées</b>	<b>61</b>
1	Introduction . . . . .	61
2	<i>Scan matching</i> par Transformée de Fourier-Mellin . . . . .	62
2.1	Introduction . . . . .	62
2.2	Principe . . . . .	63
2.2.1	Calcul d'une translation . . . . .	64
2.2.2	Rotation et homothétie . . . . .	65
2.3	Mise en œuvre et validation . . . . .	66
2.3.1	Un espace discret . . . . .	66
2.3.2	Essai sur une image artificiellement translatée . . . . .	67
2.3.3	Essai sur des panoramiques . . . . .	68
2.4	Bruit sur la corrélation de phase . . . . .	68
2.5	Corrélation . . . . .	70
2.5.1	Limites . . . . .	71
2.5.2	Conclusion . . . . .	72
2.6	Filtre anti-repliement . . . . .	73
2.7	Comparaison avec une mise en correspondance de type ICP . . . . .	73
2.7.1	Introduction . . . . .	73
2.7.2	Mise en forme des panoramiques . . . . .	74
2.7.3	Comparatif <i>scan matching</i> ICP et FMT . . . . .	75
2.8	Conclusion . . . . .	79
3	Estimation des incertitudes sur la transformation rigide . . . . .	80
3.1	Introduction . . . . .	80

3.2	Principe . . . . .	81
3.3	Test sur un signal une dimension . . . . .	82
3.3.1	Validation du procédé . . . . .	84
3.3.2	Conclusion . . . . .	85
3.4	Test sur un signal deux dimensions . . . . .	86
3.5	Données réelles . . . . .	87
3.6	Conclusion . . . . .	90
4	Algorithme RS-SLAM-FMT . . . . .	90
4.1	Introduction . . . . .	90
4.2	Algorithme général de l'EKF-SLAM . . . . .	91
4.3	RS-SLAM-FMT . . . . .	92
4.3.1	Prédiction . . . . .	93
4.3.2	Mise à jour . . . . .	94
4.3.3	Bouclage . . . . .	97
4.3.4	Carte de représentation . . . . .	98
5	Validation du procédé . . . . .	99
5.1	Introduction . . . . .	99
5.2	Expérimentation urbaine sans bouclage . . . . .	99
5.3	Expérimentation urbaine avec bouclage . . . . .	102
5.4	Conclusion . . . . .	103
<b>5</b>	<b>Expérimentations et implémentation temps réel</b>	<b>105</b>
1	Introduction . . . . .	105
2	Essais comparatifs avec d'autres algorithmes de SLAM . . . . .	106
2.1	Introduction . . . . .	106
2.2	Résultats . . . . .	107
2.3	Conclusion . . . . .	108
3	Essais en environnement naturel . . . . .	109
3.1	Introduction . . . . .	109
3.2	Cas simple avec bouclage long . . . . .	110
3.3	Trajectoire avec mouvements brusques et rapides . . . . .	113
3.4	Conclusion . . . . .	115
4	Expérimentations péri-urbaines sans objets mobiles . . . . .	118
4.1	Introduction . . . . .	118
4.2	Description . . . . .	118
4.3	Conclusion . . . . .	121
5	Expérimentations péri-urbaines avec objets mobiles . . . . .	122
5.1	Introduction . . . . .	122
5.2	Premier essai en présence d'objets mobiles . . . . .	122
5.3	Deuxième essai avec objets mobiles . . . . .	125
6	Analyse des essais . . . . .	129
6.1	Influence des capteurs proprioceptifs sur les observations . . . . .	131
7	Comparaison RS-SLAM-FMT et ROLAM . . . . .	133
7.1	Introduction . . . . .	133
7.2	Principe du ROLAM . . . . .	134
7.3	Essai comparatif . . . . .	135
7.4	Conclusion . . . . .	137
8	Implémentation temps réel . . . . .	139

8.1	Introduction . . . . .	139
8.2	Implémentation temps réel . . . . .	139
8.3	Résultats et conclusion . . . . .	141
9	Conclusion . . . . .	143
<b>Conclusions et perspectives</b>		<b>145</b>
<b>ANNEXES</b>		<b>149</b>
<b>A Algorithme général de l'EKF-SLAM</b>		<b>151</b>
1	Introduction . . . . .	151
2	Espace d'état . . . . .	151
3	Transition d'état . . . . .	152
4	Fonction d'observation . . . . .	153
5	Fonction de cartographie . . . . .	153
<b>B Essais en conditions réelles</b>		<b>155</b>
1	Introduction . . . . .	155
2	Essai A.1.3 : cas simple avec rebouclage court . . . . .	155
3	Essai A.2.2 : environnement très dense . . . . .	155
4	Essai A.3.1 : changement d'attitude . . . . .	156
5	Essai A.3.2 : influence des mouvements du robot . . . . .	158
<b>C Portage temps réel du RS-SLAM-FMT</b>		<b>163</b>
1	Introduction . . . . .	163
2	Architecture logicielle développée . . . . .	164
3	Conclusion . . . . .	166
<b>Bibliographie</b>		<b>167</b>

# Table des figures

1.1	Chariot pointant vers le sud. . . . .	3
1.2	Boussole. . . . .	4
1.3	Astrolabe. . . . .	4
1.4	Octant : mesure la hauteur d'un astre. . . . .	6
1.5	Horloge marine de Harrison. . . . .	6
1.6	Constellation de satellites GPS. . . . .	10
1.7	Système CP-DGPS. . . . .	10
1.8	Exemple de capteurs extéroceptifs. . . . .	11
1.9	Estimation de la position d'un robot. . . . .	12
1.10	Localisation d'un véhicule dans une carte connue. . . . .	12
1.11	Algorithme général du SLAM. . . . .	14
1.12	Carte de localisation et environnement réel. . . . .	15
2.1	Principe du radar FMCW pour la mesure des distances. . . . .	20
2.2	Signal radar. . . . .	22
2.3	Exemple d'acquisition radar. . . . .	23
2.4	Radar K2Pi. . . . .	24
2.5	Lobe d'antenne. . . . .	24
2.6	Signaux radar K2Pi. . . . .	24
2.7	Panoramique radar brut. . . . .	24
2.8	Radar arrêté. . . . .	26
2.9	Radar en mouvement. . . . .	26
2.10	Véhicule arrêté. . . . .	26
2.11	Véhicule en mouvement. . . . .	26
2.12	Calcul de la position des faisceaux radar. . . . .	27
2.13	Panoramique sans correction Doppler. . . . .	28
2.14	Panoramique avec correction Doppler. . . . .	28
2.15	Panoramique sans correction du cône de détection. . . . .	28
2.16	Exemple de signal convolué. . . . .	29
2.17	Exemple de signal déconvolué. . . . .	30
2.18	Panoramique avec correction du cône de détection. . . . .	30
2.19	Filtrage des panoramiques. . . . .	31
2.20	Comparaison de panoramiques. . . . .	31
3.1	Illustration de deux panoramiques successifs. . . . .	37
3.2	Cartographie par grille d'occupation [Thrun, 2000]. . . . .	39
3.3	Carte dense orientée grille d'occupation. . . . .	41
3.4	Carte topologique [Bailey, 2002]. . . . .	41

3.5	Carte à base d'amers. . . . .	43
3.6	Observation d'un amer. . . . .	44
3.7	Poses successives. . . . .	48
3.8	Graphique de nœuds. . . . .	48
3.9	Mise en correspondance de données laser. . . . .	51
3.10	Recherche des paramètres d'une droite passant par deux points. . . . .	54
3.11	Représentation dans l'espace de Hough. . . . .	55
3.12	Histogramme des angles [Röfer, 2002]. . . . .	56
3.13	Transformation rigide par la méthode des histogrammes. . . . .	58
4.1	Principe du SLAM radar. . . . .	62
4.2	Algorithme de Fourier-Mellin. . . . .	66
4.3	Transformation de $ F(w_x, w_y) $ en $F(\theta, \rho)$ . . . . .	67
4.4	Superposition automatique de deux images. . . . .	67
4.5	Pic de corrélation représentant les valeurs de translation. . . . .	68
4.6	Exemple de calcul de la transformation rigide. . . . .	68
4.7	Pic de corrélation issu de la comparaison de deux panoramiques. . . . .	69
4.8	Pic de corrélation après bruitage des images. . . . .	69
4.9	Erreur de mise en correspondance. . . . .	69
4.10	Mise en correspondance de panoramiques très différents. . . . .	70
4.11	Problèmes des parties non communes entre panoramiques. . . . .	70
4.12	Nouveau principe de calcul de la superposition. . . . .	71
4.13	Image de la corrélation. . . . .	71
4.14	Problème du manque d'informations dans les panoramiques. . . . .	72
4.15	Ambiguïtés dans le calcul des superpositions. . . . .	72
4.16	Robot équipé d'un scan laser. . . . .	74
4.17	Comparaison des détections laser et radar. . . . .	74
4.18	Histogramme du panoramique 4.17. . . . .	75
4.19	Panoramique radar après seuillage. . . . .	75
4.20	Binarisation d'un panoramique radar. . . . .	76
4.21	Transformations rigides trouvées via FMT. . . . .	76
4.22	Transformations rigides trouvées via ICP. . . . .	77
4.23	Transformations rigides trouvées via ICP, après bruitage. . . . .	78
4.24	Exemples de superposition via ICP. . . . .	79
4.25	Corrélation de phase entre signaux identiques. . . . .	81
4.26	Corrélation de phase entre signaux légèrement différents. . . . .	81
4.27	Corrélation de phase entre signaux très différents. . . . .	82
4.28	Test sur signal radar une dimension $X_1$ . . . . .	82
4.29	Bruitage de $X_2$ . . . . .	83
4.30	Résultat de la corrélation de phase entre $X_1$ et $X_2$ . . . . .	83
4.31	Fenêtrage du signal de corrélation. . . . .	84
4.32	Calcul de 800 translations. . . . .	85
4.33	Simulation de deux panoramiques. . . . .	86
4.34	Fenêtrage de l'image de corrélation. . . . .	86
4.35	Calcul de 80 translations. . . . .	87
4.36	Zone de test. En rouge la vérité terrain issue d'un GPS-RTK. . . . .	88
4.37	Erreurs en $X$ sur données réelles. . . . .	88
4.38	Erreurs en $Y$ sur données réelles. . . . .	89

4.39	Erreurs en angle sur données réelles. . . . .	89
4.40	Description de la zone d'acquisition. . . . .	90
4.41	Les différentes étapes de l'algorithme EKF-SLAM. . . . .	91
4.42	Modèle d'évolution du véhicule (modèle point). . . . .	93
4.43	SLAM dans un environnement péri-urbain. . . . .	95
4.44	Exemples de panoramiques radar. . . . .	95
4.45	Choix des Mises en correspondances. . . . .	96
4.46	Principe du bouclage. . . . .	98
4.47	Parcours péri-urbain de 1135 m. . . . .	100
4.48	Estimation du parcours par l'algorithme <i>RS-SLAM-FMT</i> . . . . .	100
4.49	Carte reconstruite à partir des positions estimées. . . . .	101
4.50	Exemple d'un panoramique issu du radar K2Pi. . . . .	101
4.51	Parcours bouclé de 700 m. . . . .	102
4.52	Estimation de parcours avant bouclage. . . . .	103
4.53	Estimation de parcours après bouclage. . . . .	103
4.54	Carte construite à partir des positions estimées. . . . .	104
5.1	Trajectoires estimées par les trois techniques. . . . .	107
5.2	Erreurs sur les positions estimées. . . . .	108
5.3	Moyenne des erreurs. . . . .	109
5.4	Photographie du R-Trooper. . . . .	110
5.5	Vue aérienne du parcours. . . . .	110
5.6	Tracés comparatifs du parcours « bouclage long » . . . . .	112
5.7	Erreurs entre la vérité terrain et le parcours estimé. . . . .	113
5.8	Cartographie dense de la zone d'exploration. . . . .	114
5.9	Vue aérienne du parcours. . . . .	115
5.10	Estimation du parcours par une simple prédiction. . . . .	116
5.11	Tracés comparatifs du parcours « mouvements brusques » . . . . .	116
5.12	Erreurs entre la vérité terrain et le parcours estimé. . . . .	117
5.13	Cartographie dense de la zone d'exploration. . . . .	117
5.14	Velac : véhicule de test du LASMEA. . . . .	118
5.15	Vue aérienne du parcours. . . . .	119
5.16	Estimation du parcours. . . . .	120
5.17	Erreurs entre la vérité terrain et le parcours estimé. . . . .	120
5.18	Cartographie dense de la zone d'exploration. . . . .	121
5.19	Superposition d'une vue aérienne et de la cartographie dense. . . . .	122
5.20	Vue aérienne du parcours. . . . .	123
5.21	Photographies des zones traversées. . . . .	124
5.22	Résultats du SLAM en conditions réelles. . . . .	125
5.23	Cartographie dense de la zone d'exploration. . . . .	126
5.24	Vue aérienne du parcours. . . . .	127
5.25	Photographies des zones traversées. . . . .	128
5.26	Résultats du SLAM en conditions réelles. . . . .	129
5.27	Cartographie dense de la zone d'exploration. . . . .	130
5.28	Parcours simulé avec un environnement type péri-urbain. . . . .	131
5.29	Distribution des erreurs. Sans bruits. . . . .	132
5.30	Distribution des erreurs, avec du bruit sur $\theta$ . . . . .	132
5.31	Distribution des erreurs, avec du bruit sur l'odométrie. . . . .	133

5.32	Distribution des erreurs, avec du bruit sur l'odométrie et sur $\theta$ . . . . .	133
5.33	Principe du ROLAM. . . . .	134
5.34	Position d'un faisceau radar lors d'un déplacement en ligne droite. . .	135
5.35	Description du procédé ROLAM. . . . .	135
5.36	Trajectoires estimées par le ROLAM et le RS-SLAMT-FMT. . . . .	136
5.37	Erreurs en distance. . . . .	137
5.38	Vue aérienne de la zone d'expérimentation. . . . .	137
5.39	Cartographie obtenue par le RS-SLAM-FMT. . . . .	138
5.40	Cartographie obtenue par le ROLAM [Vivet et al., 2012]. . . . .	138
5.41	Interface fonctionnelle. . . . .	140
5.42	Zone urbaine traversée par le R-Trooper. . . . .	141
5.43	RS-SLAM-FMT temps réel. . . . .	142
B.1	Vue aérienne du parcours. . . . .	157
B.2	Parcours estimé par le RS-SLAM-FMT. . . . .	158
B.3	Vue aérienne du parcours. . . . .	158
B.4	Parcours estimé par le RS-SLAM-FMT. . . . .	159
B.5	Vue aérienne du parcours. . . . .	160
B.6	Parcours estimé par le RS-SLAM-FMT. . . . .	161
B.7	Vue aérienne du parcours. . . . .	162
B.8	Parcours estimé par le RS-SLAM-FMT. . . . .	162
C.1	Architecture générale RTMaps. . . . .	163
C.2	Distribution des tâches sous RTMaps. . . . .	164
C.3	Représentation temporelle des signaux radars et proprioceptifs. . . . .	164
C.4	Architecture logicielle mise en place. . . . .	165

# Liste des tableaux

2.1	Caractéristiques du radar FMCW K2Pi. . . . .	23
5.1	Caractéristiques de l'essai de bouclage long . . . . .	111
5.2	Caractéristiques de l'essai « mouvements brusques » . . . . .	113
5.3	Caractéristiques de l'essai « zenith ». . . . .	119
5.4	Caractéristiques du premier essai en conditions réelles. . . . .	123
5.5	Caractéristiques du deuxième essai en conditions réelles. . . . .	126
5.6	Temps de calcul des deux processus principaux. . . . .	142
B.1	Liste des essais effectués en environnement naturel. . . . .	156
B.2	Caractéristiques de l'essai A.1.3. . . . .	157
B.3	Caractéristiques de l'essai A.2.2. . . . .	159
B.4	Caractéristiques de l'essai A.3.1. . . . .	160
B.5	Caractéristiques de l'essai A.3.2. . . . .	161
C.1	Caractéristiques des processus. . . . .	166





# Chapitre 1

## Introduction

Le système GPS « *Global Positioning System* » utilise une localisation par satellite qui permet de fournir la position globale de n'importe quel point sur la planète. Si on l'associe à une carte connue, on peut considérer que c'est l'outil idéal pour positionner et orienter un robot mobile dans un milieu extérieur.

Il faut néanmoins considérer que plusieurs problèmes peuvent venir troubler cette localisation tels que la mauvaise réception satellitaire, les problèmes de réflexions et de brouillage, les interventions dans une zone non couverte, etc. De plus, la connaissance d'une carte *a priori* suppose qu'aucune modification de l'environnement ne puisse avoir lieu. Il est donc nécessaire d'utiliser d'autres méthodes de localisation en complément.

Le SLAM, « *Simultaneous Localisation And Mapping* » représente à l'heure actuelle l'une des principales thématiques investiguées dans le domaine des robots mobiles autonomes. Il permet, à l'aide de capteurs extéroceptifs<sup>1</sup> (laser, caméra, radar, etc.) et proprioceptifs<sup>2</sup> (odomètre, gyromètre, etc.), de trouver l'orientation et la localisation d'un robot dans un environnement extérieur vaste, inconnu ou modifié.

Les travaux décrits dans ce manuscrit s'intègrent dans ce courant de recherche et visent à développer un SLAM innovant qui utilise un radar à modulation de fréquence continue « *FMCW* » comme capteur extéroceptif dans le cadre de la localisation temps réel d'un véhicule d'exploration. Ces travaux s'inscrivent dans le cadre d'un projet ANR intitulé « **IMPALA**, Radar panoramique hyperfréquence pour la localisation et la cartographie simultanées en environnement extérieur ». Il est mené en partenariat avec l'Institut de recherche pour l'ingénierie de l'agriculture et de l'environnement « *Cemagref* » qui développe le radar, le Laboratoire des Sciences et Matériaux pour l'Electronique et l'Automatique « *LASMEA* » qui développe les algorithmes de localisation et THALES TOSA<sup>3</sup> qui teste et valide le procédé sur un véhicule d'exploration.

L'utilisation d'un radar va permettre de s'affranchir de certaines limites connues des capteurs usuels. Elles sont généralement liées aux conditions climatiques, au type d'environnement, à la vitesse du véhicule, et à la faible portée de détection. Les caractéristiques intrinsèques du capteur semblent idéales. Il faut néanmoins pondérer cela par le fait qu'elles fournissent des informations fortement bruitées et qui

---

1. Système de mesure qui fournit l'état de l'environnement du véhicule.

2. Système de mesure qui fournit l'état interne du véhicule.

3. THALES optronique S.A.

manquent de précision. La littérature présente quelques cas d'utilisations de ce type de capteurs dans des algorithmes de SLAM. Ils restent rares et peu adaptés à des conditions d'utilisation péri-urbaines (alternance d'immeubles, de zones naturelles, de routes, et avec la présence ou non d'objets mobiles).

Les travaux décrits dans cette thèse proposent un système qui fonctionne en temps réel dans des environnements variés pour en réaliser l'exploration et la cartographie dense et précise. Pour ce faire, il utilise une technique de SLAM par « approche directe » comme l'ont proposé pour la première fois Lu et Milios [1994] dans les années 90. Cette méthode permet notamment de s'affranchir des problèmes liés à l'extraction d'amers<sup>4</sup>. Adaptée au radar, cette approche permet de calculer les déplacements du véhicule dans la zone explorée. Pour la première fois, dans la littérature SLAM, ces calculs sont effectués avec un algorithme de mise en correspondance qui utilise les propriétés de la transformation de Fourier-Mellin (*FMT*).

Le système, quelles que soient ses performances, doit fournir à l'utilisateur des informations d'incertitudes sur les localisations estimées. Cela fait partie intégrante des algorithmes de SLAM et nécessite une connaissance précise des incertitudes de mesures. Les travaux proposés vont présenter un nouveau procédé de détermination des incertitudes directement associé à la méthode de mise en correspondance *FMT*.

La suite de ce chapitre donne une introduction générale sur le rôle de la navigation dans la civilisation humaine. Il se poursuit par une définition des outils de la navigation moderne et des principes associés. Il se termine par une description de l'organisation du manuscrit.

## 1 Bref historique des outils de navigation

Depuis son origine, l'homme cherche par tous les moyens à se repérer dans son milieu. Sa soif d'exploration l'amène à se déplacer sans cesse avec pour objectif la découverte de nouvelles contrées, illustrée de nos jours par la conquête spatiale.

A ses débuts, en tant qu'explorateur, son ingéniosité lui permit d'utiliser diverses informations pour tenter de se repérer : les amplitudes des marées, la forme des côtes, les cycles lunaires, l'orientation et la périodicité des vents, ainsi que l'observation du soleil et des étoiles.

Que dire de ces Vikings qui découvrirent l'Amérique cinq siècles avant Christophe Colomb avec comme seul outil l'observation des côtes et des étoiles.

C'était le début d'une nouvelle science, la navigation. Elle a pour objet principal la résolution de deux problèmes :

- déterminer la route à suivre pour se rendre d'un lieu à un autre,
- faire le point c'est-à-dire déterminer sa position.

Les premières grandes expéditions se firent sur les mers. Elles s'appuyèrent sur deux principes :

- la navigation à l'estime dans laquelle les éléments des calculs sont tirés uniquement des indications du loch<sup>5</sup> et du compas,

---

4. Points remarquables présents dans l'environnement

5. Le loch est un instrument de navigation maritime qui permet d'estimer la vitesse de déplacement d'un navire sur l'eau, c'est-à-dire sa vitesse relative en surface par rapport à la masse d'eau

- la navigation observée dans laquelle on utilise des observations d’objets de positions connues. Elle se subdivise en navigation côtière et en navigation astronomique suivant que les objets observés sont des points terrestres ou des astres.

Une recherche sur Wikipédia donne un rapide résumé sur les principes de la navigation maritime et les outils de mesure associés. En voici quelques extraits :

comment ne pas citer cette invention chinoise de 2600 av. J.-C., « *le chariot pointant vers le sud* » (cf. figure 1.1), ancêtre de la boussole magnétique : c’est un dispositif mécanique qui permet de compenser les rotations du véhicule afin qu’une partie déterminée du chariot pointe constamment vers le sud.



FIGURE 1.1 – Chariot pointant vers le sud.

Les innovations humaines, liées à des siècles d’observations, permirent d’améliorer sans cesse les routes et cartes de navigation. Si on remonte aux premiers pas de la boussole (cf. figure 1.2) qui datent du  $XII^e$  siècle<sup>6</sup>.

On peut associer les grandes découvertes géographiques à des avancées techniques majeures :

- au  $IV^e$  siècle av. J.-C., les phéniciens sont les maîtres de la méditerranée. Naviguer consiste à caboter, c’est-à-dire suivre de près le littoral, d’escale en escale, sans perdre la côte de vue. On se guide au moyen des étoiles avec un empirisme qui se transmet d’un marin à l’autre ;
- jusqu’au  $III^e$  siècle, les grecs puis les romains poursuivent les explorations « proches », ils complètent l’élaboration des cartes ;

---

où il évolue.

6. On s’accorde aujourd’hui pour considérer que ce furent les chinois qui découvrirent la faculté qu’a une tige métallique aimantée de toujours pointer vers une même direction, ceci bien avant notre ère.



FIGURE 1.2 – Boussole.

- au moyen âge les vikings deviennent les premiers navigateurs de haute mer, en utilisant simplement l’observation des étoiles et des côtes ;
- au  $XIII^e$  siècle, la boussole fait son apparition en Méditerranée, on dispose alors des informations de vitesse (loch) et de cap pour réaliser de la navigation à l’estime ;
- au  $XIV^e$  siècle, la latitude est obtenue en mesurant la hauteur de l’étoile polaire avec l’astrolabe<sup>7</sup> (cf. figure 1.3) nautique, puis avec l’anneau astronomique, l’arbalestrille<sup>8</sup>, le quart nautique. On peut donc déterminer son éloignement par rapport à l’équateur ;

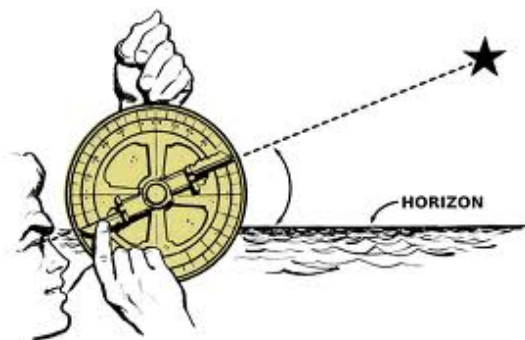


FIGURE 1.3 – Astrolabe.

- en 1471, des navigateurs portugais franchissent l’équateur ;
- en 1486, le portugais Diaz, mandaté pour trouver une route vers l’Inde loin des côtes, contourne l’Afrique par le sud et découvre le Cap de Bonne Espérance. C’est le premier exemple de navigation en haute mer ;
- en 1492, inspiré par Diaz, Christophe Colomb cherche à rejoindre les Indes par l’ouest, il découvre en fait les Antilles ;

---

7. Instrument pour prendre la hauteur des astres.

8. Instrument de mesure permettant de mesurer l’angle entre l’horizon et un astre.

- en 1498, le portugais Vasco de Gama atteint les Indes à Calicut, après avoir accompli le tour de l’Afrique ;
- en 1499, le pilote florentin Amerigo Vespucci, qui fut au service de l’Espagne et du Portugal, propose l’observation des occultations d’étoiles pour la détermination des longitudes. Il a été, semble-t-il, le premier à avoir eu l’idée de la méthode des distances lunaires<sup>9</sup> pour cette même détermination. Ce sont les premiers pas vers une localisation absolue.

Ces premiers pas dans la navigation hauturière<sup>10</sup> se font à l’estime en partie rectifiée par l’observation de la latitude. Se pose alors un problème vital pour les marins, celui de savoir exactement sa position en mer (loin des côtes il n’y a plus repères), c’est-à-dire la latitude (donne la position nord-sud) et la longitude (donne la position est-ouest) du point où il se trouve.

Pour mesurer la longitude, il faut calculer (en admettant que les 360° de circonférence de la terre correspondent à 24 heures, sur la base de 15° = 1 heure) la différence entre l’heure locale et l’heure du lieu d’origine d’un bateau. Ce résultat permet de déterminer la longitude grâce à des méthodes astrologiques. Si on peut observer un phénomène astronomique régulier à bord d’un bateau, l’heure de cette observation peut ensuite être comparée avec l’heure d’observation du même phénomène au point de départ du bateau. On peut trouver ces données sur des tables dressées à cet usage. La différence des deux temps donne la longitude.

Pour calculer le temps local, les astronomes vont tenter de mettre au point la méthode de la distance lunaire. Le principe est d’utiliser les mouvements de la lune par rapport au reste des astres et d’utiliser ces mouvements comme une horloge. Les astronomes travaillent donc sur la conception de tables horaires du trajet de la lune sur le fond étoilé. Un travail de longue haleine tant les mouvements de la lune sont complexes. Cette méthode astronomique sera finalement supplantée par la simplicité et la précision d’une petite horloge.

- En 1596, le navigateur hollandais Barentz essaie de se servir de montres pour déterminer la longitude selon les principes de la navigation céleste. La mauvaise précision de l’horloge provoque des erreurs importantes ;
- en 1731, création par Hadley de l’octant (cf. figure 1.4) qui allait ouvrir l’ère des distances lunaires pour la mesure du temps ;
- en 1736, première « horloge marine à longitude » de John Harrison (cf. figure 1.5). Sa précision permet enfin une détermination correcte de la longitude.

Depuis la fin du *XVIII*<sup>e</sup> siècle la méthode générale pour déterminer le point à la mer avec les chronomètres consiste à obtenir la latitude par une hauteur méridienne de soleil et la longitude par une hauteur horaire (ancienne méthode du Point à midi). La deuxième moitié du *XIX*<sup>e</sup> siècle voit le triomphe des chronomètres sur les distances lunaires pour la détermination des longitudes.

La navigation à l’estime est complétée par une navigation absolue dans laquelle on peut déterminer à chaque instant sa position globale (sous condition de pouvoir observer les astres).

---

9. Détermination de la longitude par la mesure d’une distance angulaire entre la lune et un astre.

10. Cela représente la navigation en haute mer, avec la recherche du chemin le plus court pour se déplacer entre deux points.



FIGURE 1.4 – Octant : mesure la hauteur d'un astre.



FIGURE 1.5 – Horloge marine de Harrison.

## 2 Les outils de la navigation moderne

Les premiers pas de la navigation moderne se sont faits par l'utilisation des ondes radio, par la découverte des antennes à cadre directionnel, et par le développement de la radiogoniométrie<sup>11</sup> associée.

Le guidage des navires en Atlantique Nord par des radiophares fut l'une des premières applications.

En mesurant la direction de deux radiophares avec un radiocompas, le navigateur détermine un point probable par triangulation. Ces radiocompas se sont perfectionnés jusqu'aux modèles actuels encore largement utilisés en aéronautique. Le système *CONSOL*<sup>12</sup> est un perfectionnement développé pendant la seconde guerre mondiale qui ne demande pas d'antenne directive embarquée.

Une évolution majeure a été obtenue entre 1940 et 1945 avec les premiers systèmes hyperboliques, le *LORAN*<sup>13</sup> et le *DECCA*<sup>14</sup>. Les systèmes hyperboliques déterminent la position en mesurant la différence de temps de propagation entre deux émetteurs (au minimum), le lieu des points à différence égale étant représenté par une hyperbole sur la carte. Trois émetteurs sont nécessaires pour un point (intersection d'hyperboles). Pour éviter des géométries imprécises ou ambiguës, on augmente le nombre d'émetteurs.

Le premier système opérationnel par satellite est le *TRANSIT*<sup>15</sup>. La position était calculée à partir de mesures successives de l'effet Doppler sur les signaux émis par le satellite qui diffusait en parallèle des éphémérides, permettant le calcul de sa position (exprimée dans le système géodésique WGS 72). La précision de la position était généralement inférieure à 1 km. Le système de navigation par satellite *TRANSIT* est arrêté depuis le 31 décembre 1996.

Les systèmes actuels utilisent :

1. les systèmes de positionnement par satellite. Ils permettent à un récepteur mo-

11. Détermination de la direction d'arrivée d'une onde électromagnétique.

12. Invention allemande portant le nom de *SONNE*.

13. LONg RANge Navigation system.

14. Du nom de la société qui le développa durant la seconde guerre mondiale : Decca Radio LTD.

15. Système de navigation par satellites mis au point pour la marine des États-Unis par l'université Johns-Hopkins à partir de 1958 et mis en œuvre pour la première fois en 1964. Il fallu attendre 1967 pour une application civile.

bile de se positionner dans un système géodésique, à l'aide d'une constellation de satellites en orbite. On peut citer *GPS*<sup>16</sup> (américain), *GLONASS*<sup>17</sup> (russe), et dans le futur, *GALILEO*<sup>18</sup> (européen) et *BEIDOU*<sup>19</sup> (chinois) ;

2. les systèmes à base terrestre. Ces systèmes utilisent des balises fixes (radio-phares, radiobalises) pour permettre à un récepteur mobile de se positionner :
  - les systèmes côtiers : le *SYLEDIS*<sup>20</sup> en Europe occidentale et les radio-phares ;
  - les systèmes hauturiers : le *LORAN C* (Atlantique nord, Pacifique nord, Océan Indien nord), le *LORAN A* (Chine, Japon) et *Chaïka* (Pacifique nord-ouest) ;
  - les systèmes aéronautiques : radiobalises, *VOR*<sup>21</sup>, *ILS*<sup>22</sup> et *DME*<sup>23</sup>.

### 3 Limites du positionnement global

Que pouvait faire un capitaine du *XVIII<sup>e</sup>* siècle, lorsque le ciel était très nuageux et qu'il ne pouvait pas utiliser son sextant pour déterminer sa position ?

Il pouvait prédire ses déplacements en partant de sa dernière position connue et en modélisant sur une carte, à l'aide des informations fournies par son loch et son compas, ses déplacements. Il pouvait compléter cela en tentant d'observer des points remarquables (amers) représentés sur la carte.

Que peut faire l'homme moderne lorsqu'il est dans son véhicule et que le positionnement GPS est inopérant ?

A partir de sa dernière position connue, il peut estimer ses déplacements sur une carte en prenant en compte la distance qu'il accomplit dans une direction donnée. Il peut compléter cela en observant le monde qui l'entoure afin d'en extraire des points d'intérêt également présents sur la carte (intersections, bâtiments, croix ...). On parle dans la littérature de primitives (*features* en anglais) ou d'amers (*landmarks* en anglais). Ce sont des parties distinctives de l'environnement que l'on peut facilement extraire via un capteur extéroceptif (cf. section 5.1.2).

### 4 De l'homme au robot autonome

Depuis plusieurs décennies l'homme cherche à créer des véhicules autonomes afin de le remplacer lors d'interventions dans des zones dangereuses.

Ces modules susceptibles de se déplacer seuls et sans risques doivent intégrer de multiples processus dont celui assurant la navigation. Dans ce contexte, les considé-

---

16. Global Positioning System.

17. Signifie dans sa traduction française : système global de navigation satellitaire.

18. Son nom est un hommage au savant Italien Galileo Galilei. Ce système sera opérationnel en 2014.

19. Aussi connu sous le nom : Compass Navigation Satellite System, il vise à s'affranchir de la dépendance au GPS américain. Il devrait être opérationnel en 2013.

20. Système léger de mesure de distance.

21. Vhf Omnidirectional Range.

22. Instrument Landing System.

23. Distance Measuring Equipment.



rations humaines précédemment décrites peuvent être reprises : « la navigation c'est être capable de se localiser et de s'orienter ».

Pour cela on doit disposer d'informations :

- proprioceptives : pour les premiers navigateurs elles étaient issues du loch (mesure de vitesse) et du compas. Elles sont dans le cadre d'un véhicule moderne issues d'un odomètre, d'un gyromètre, etc. ;
- extéroceptives : fournissent des observations sur l'environnement proche. Ce fut l'observation par les marins des étoiles, des côtes, des phares (premier amer artificiel). A l'heure actuelle, ce sont des caméras, des lasers, des sonars, des radars ;
- absolues : elles vont permettre de positionner le robot dans un repère global. Cela correspond à la détermination de la latitude et de la longitude par GPS ou par des méthodes plus anciennes ;
- cartographiques : on peut vouloir déplacer le robot dans un environnement référencé, on utilise alors une carte connue. Dans le cadre de l'exploration on peut utiliser le robot pour créer cette carte au fur et à mesure de l'exploration.

## 5 Localisation autonome et cartographie

De nos jours, la navigation par satellite fait partie de notre quotidien. Dans un futur proche, associée à des capteurs « de proximité » proprioceptifs et extéroceptifs, elle va permettre à tout système autonome d'avoir une fiabilité de navigation optimale permettant ainsi des actions en totale autonomie c'est-à-dire sans aucune intervention humaine.

Il faut néanmoins pondérer ces affirmations par le fait que l'on ne peut pas toujours garantir la couverture GPS, que cette dernière n'est pas toujours précise et que l'on ne possède pas forcément de cartographie *a priori*. Dans ce cas on doit pouvoir se localiser et construire une carte par l'utilisation des seuls capteurs de « proximité ».

Au quotidien, un véhicule autonome pourra rencontrer deux cas de figures en fonction des informations dont il disposera :

- la navigation en zone connue, dans laquelle on utilise une cartographie existante. Il suffit au robot de se localiser (via le GPS par exemple) dans cette carte puis de décider de son orientation future (connaissant son orientation) en fonction de la destination finale à atteindre ;
- la navigation exploratrice, dans ce cas la zone est inconnue (donc sans cartographie référencée) : le robot se déplace et découvre la zone. Il doit, à l'aide de ses capteurs, construire au fur et à mesure de son trajet sa propre carte et définir sa position dans cette carte. On appelle cela du *SLAM* (Simultaneous Localisation And Mapping).

La problématique générale est fonction des capteurs utilisés, des zones traversées, de la cartographie connue ou non, et des méthodes mathématiques choisies.

### 5.1 Capteurs

Une grande variété de capteurs peut être utilisée en navigation. On classe traditionnellement les capteurs en deux catégories selon qu'ils mesurent l'état du mobile

lui-même (les capteurs proprioceptifs) ou l'état de son environnement (les capteurs extéroceptifs).

### 5.1.1 Les capteurs proprioceptifs

On peut citer quelques exemples de capteurs proprioceptifs : la centrale inertielle, le gyromètre, l'odomètre, le compteur, etc. Ils fournissent des mesures qui correspondent à des variations de position, d'orientation, et d'incrément. Une centrale inertielle donne des informations d'accélération, un gyromètre renseigne sur la vitesse angulaire, et un odomètre sur le cumul des distances parcourues.

Les informations provenant de ces capteurs sont généralement utilisées avec un modèle mathématique du véhicule afin de pouvoir prédire la trajectoire de ce dernier. Il y a une limitation intrinsèque à l'utilisation de cette technique. En effet, les bruits de mesure vont se cumuler à chaque prédiction calculée à partir du modèle du véhicule, fournissant à terme une estimation de position fortement bruitée. En pratique, une estimation de position seulement fondée sur ce principe est irréalisable lors de longs parcours (on peut exclure de cette règle certaines centrales inertielles très onéreuses qui possèdent un bruit de mesure très faible).

Ces capteurs possèdent néanmoins certaines caractéristiques intéressantes qui en font des outils précieux pour des mesures instantanées (et donc des prédictions estimatives sans cumul) :

- une fréquence d'échantillonnage élevée ;
- une passivité (pas d'émission d'ondes électromagnétiques ou autres) ;
- une indépendance vis à vis de l'environnement extérieur. Un gyromètre fonctionne aussi bien sur terre, dans l'air ou sous les mers.

### 5.1.2 Les capteurs extéroceptifs

Ils peuvent se classer en deux catégories. Il y a ceux qui fournissent une position absolue dans le référentiel terrestre (le GPS) et ceux qui fournissent une position et/ou une orientation locale dans une zone de détection donnée.

#### Le GPS

Le positionnement par satellite est rendu possible grâce à une constellation de 24 satellites qui décrivent chaque jour la même orbite. Entre 5 et 8 satellites sont toujours visibles de n'importe quel point de la terre (cf. figure 1.6) ce qui permet d'obtenir une bonne précision du positionnement par triangulation.

La précision de l'ordre du centimètre obtenue par les CP-DGPS<sup>24</sup> permet de les utiliser comme capteur en vue de la commande du véhicule (cf. figure 1.7).

Le GPS est insensible aux conditions atmosphériques. Cependant ses limitations sont liées aux problèmes de qualité de couverture de la zone (lors de déplacements dans des enceintes fermées, dans des zones géographiques très enclavées, etc.), au brouillage des informations (lors d'interventions militaires), aux réflexions multiples liées à l'infrastructure, et à l'existence de domaines non couverts (exploration spatiale).

Le GPS est classé dans la catégorie des capteurs de positionnement absolu, en marge des autres capteurs extéroceptifs.

---

<sup>24</sup> Carrier-Phase Differential Global Positioning System

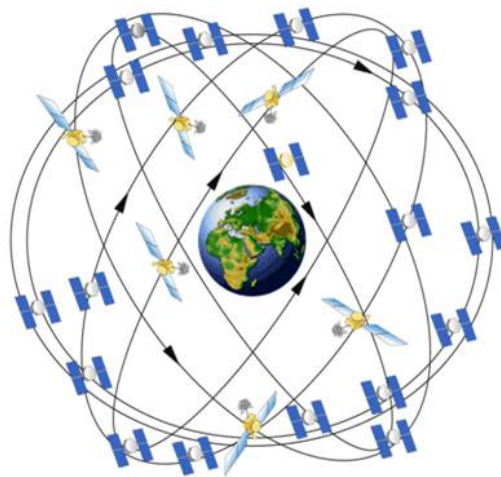


FIGURE 1.6 – Constellation de satellites GPS détectés par un véhicule terrestre (<http://www.gps.gov>).

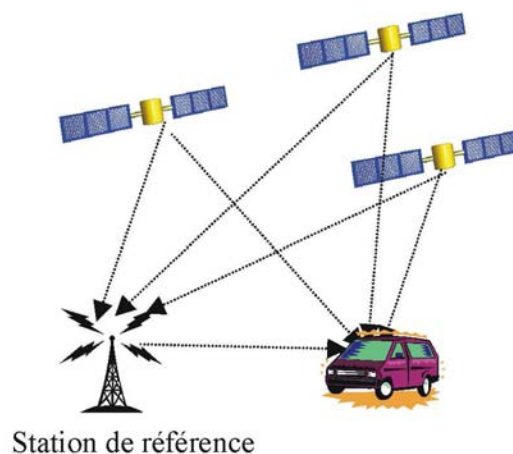


FIGURE 1.7 – Système CP-DGPS.

### Les capteurs à détection proche

Les capteurs de type radar, sonar, télémètre laser sont des capteurs dits actifs (cf. figure 1.8). Ils sont composés d'un émetteur qui envoie un rayonnement, et d'un récepteur qui le reçoit (après réflexion par la scène observée). La comparaison entre l'onde émise et l'onde reçue permet d'établir une information de profondeur qui est issue d'une mesure de temps de vol ou de déphasage.

Les scanners laser 2D/3D fournissent les distances aux objets les plus proches et les orientations associées. Le sonar et radar fournissent également ces deux informations mais avec des précisions angulaires moindres et un bruit plus important.

Les capteurs de vision sont des capteurs « naturels » pour les problèmes d'aide à la navigation. En effet, l'essentiel des informations perçues par un conducteur sur son environnement provient de sa vision. Elle lui permet d'assurer les principales tâches de perception requises à la navigation. Deux types de capteurs de vision peuvent



FIGURE 1.8 – Exemple de capteurs extéroceptifs : la famille des télémètres lasers Sick.

être globalement distingués. Le premier est la vision stéréoscopique, c'est-à-dire la combinaison de deux ou plusieurs caméras observant la même scène. Le second est l'utilisation d'une caméra unique. Ces deux techniques, via des méthodes de traitements différentes, peuvent fournir des informations de distance et d'orientation des objets observés.

Tous ces capteurs ne fournissent pas d'informations absolues de positionnement mais des informations locales sur les positions des objets présents dans l'environnement du capteur.

## 5.2 Localisation

### 5.2.1 Localisation à l'estime

La localisation à l'estime est une méthode qui consiste à déterminer la position d'un véhicule (terrestre, maritime, aérien ou spatial ; piloté ou automatique) à partir de ses déplacements successifs (mesurés par des capteurs proprioceptifs) depuis sa dernière position connue. Traditionnellement, cette méthode repose sur des instruments mesurant le cap (compas) et la vitesse (tachymètre).

On peut prendre l'exemple de la figure 1.9 où un robot se déplace de la position réelle  $x_{R,1}$  jusqu'à la position  $x_{R,2}$ .  $\theta_{i,i+1}$  est la variation d'angle mesurée par son gyromètre entre les positions  $i$  et  $i+1$ , et  $D_{1,2}$  est la distance parcourue mesurée par son odomètre entre les positions  $i$  et  $i+1$ . On peut alors estimer la position  $x_{E,i+1}$  connaissant la position  $x_{E,i}$ .

La position estimée est incertaine car dépendante de la précision de la mesure de la route vraie. C'est la forme la plus basique de localisation. Elle détermine la position du véhicule par une simple intégration de ses mouvements dans un modèle d'évolution. L'inconvénient est que chaque estimation va utiliser des mesures bruitées. Ces erreurs accumulées vont faire que l'incertitude sur la position ne va que s'accroître avec le temps. Cette méthode ne peut donc pas servir de mécanisme de localisation sur les longs parcours. Elle restera un complément à la localisation par cartographie.

### 5.2.2 Localisation absolue avec une carte *a priori*

La localisation absolue utilise des capteurs extéroceptifs qui donne des informations sur l'environnement. La solution la plus simple est l'utilisation d'un capteur

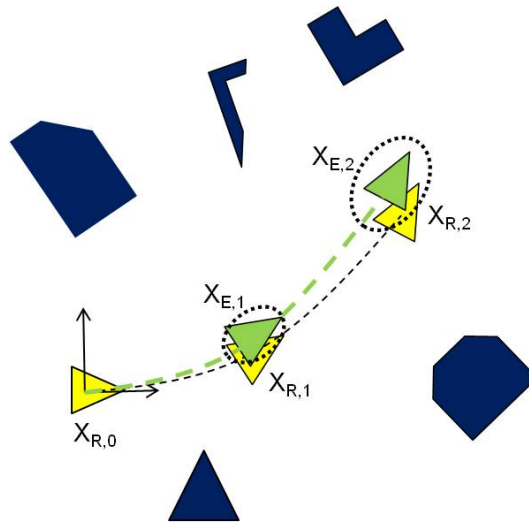


FIGURE 1.9 – Cette figure représente l'estimation de la position d'un robot (en vert) connaissant le cap et la distance parcourue. Les ellipses montrent l'incertitude sur les positions estimées.

type GPS qui indique directement la position sur une carte connue. Si l'utilisation d'un GPS n'est pas possible, le véhicule peut utiliser des capteurs du type scanner laser pour repérer des amers dans la zone parcourue. Il peut ensuite déterminer sa position par la mise en correspondance de ces amers avec une carte connue. Le problème de cette méthode réside dans le fait qu'il faut forcément avoir déjà traversé cette zone pour disposer d'une carte. Il faut également avoir repéré préalablement tous les amers avant que le véhicule autonome ne vienne s'y repérer quitte à rajouter des amers artificiels dans les zones sans reliefs. Que se passe-t-il si l'environnement change ou si on dépasse les limites de la carte ?

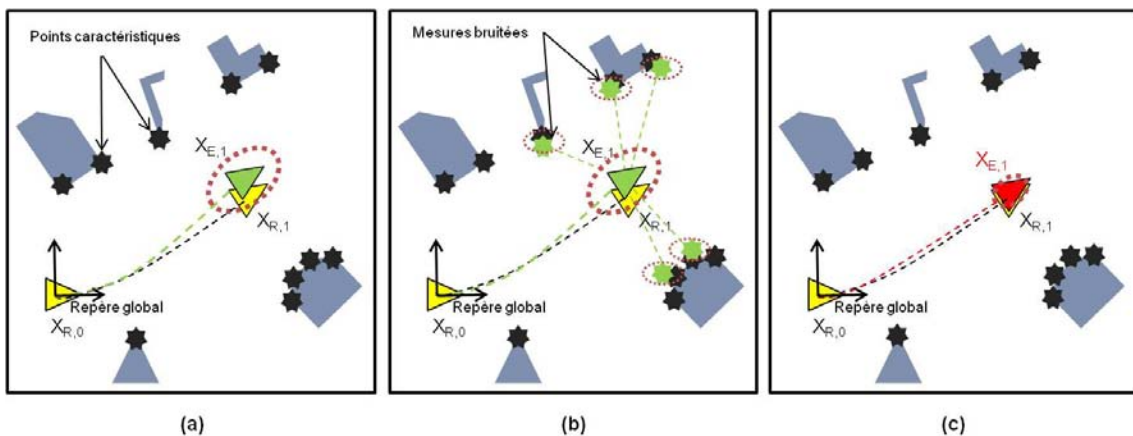


FIGURE 1.10 – Localisation d'un véhicule dans une carte connue de l'environnement : **(a)** prédiction de la position du véhicule, **(b)** acquisition de mesures sur l'environnement et **(c)** estimation de la position après recalage.

On peut résoudre le problème de la localisation par une méthode d'estimation

réursive. La figure 1.10 présente les différentes étapes effectuées à chaque itération. Le véhicule (position réelle en jaune) évolue dans un environnement connu *a priori* représenté par des étoiles grises. La localisation est réalisée en trois étapes :

- *prédiction de la position(a)* : c'est une localisation à l'estime à partir de capteurs proprioceptifs. L'incertitude sur la position du véhicule est importante (ellipse en pointillée) ;
- *acquisition de mesures sur l'environnement(b)* : des capteurs extéroceptifs mesurent la position d'objets remarquables (étoiles vertes) dans l'environnement proche du véhicule. Chacune de ces mesures est bruitée. Elle est associée à une ellipse d'incertitude. Cela forme une carte locale de l'environnement. La mise en correspondance de cette carte avec la carte connue *a priori* (étoiles grises) permet d'obtenir une correction de la localisation du véhicule ;
- *estimation finale de la position(c)* : une nouvelle estimation de la position est obtenue (en rouge) en utilisant la correction calculée en (b).

### 5.2.3 Localisation absolue sans carte *a priori*

On voit clairement apparaître les motivations des algorithmes de type SLAM qui vont se passer de la connaissance d'une carte *a priori* tout en conservant des incertitudes contenues sur le positionnement. Le SLAM va utiliser la position estimée du véhicule pour localiser des amers. Dans le même temps ces amers serviront à améliorer l'estimation de la position du véhicule. Si le capteur est de bonne qualité et l'association sans faille, les multiples observations des amers permettent de diminuer l'incertitude sur leurs positions et la qualité de la carte reconstruite augmente. Le SLAM est donc capable de faire naviguer un véhicule dans une zone inconnue, de lui faire explorer la zone afin de construire une carte *a priori* et de le ramener à son point de départ une fois le travail accompli.

La résolution du problème de SLAM implique donc la réalisation itérative de quatre étapes distinctes (cf. figure 1.11) :

- *phase initiale (a)* : le système possède une estimation de sa position (véhicule en rouge) et des points caractéristiques de l'environnement (étoiles rouges). Tous ces éléments sont évalués avec une incertitude (ellipses rouges) due aux bruits sur les mesures proprioceptives et extéroceptives ;
- *phase prédictive (b)* : le véhicule se déplace. Ses capteurs proprioceptifs fournissent des mesures permettant d'estimer sa position. Comme ces mesures sont entachées de bruits, l'incertitude sur la position du véhicule augmente ;
- *phase d'acquisition et d'association (c)* : une carte locale de l'environnement est construite à partir des mesures extéroceptives (étoiles vertes). Après mise en correspondance de cette carte avec la carte globale (association des données représentant le même point), on calcule les valeurs de correction à apporter ;
- *phase d'estimation de la cartographie (d)* : on intègre la correction dans le système afin de déterminer une nouvelle estimation des positions du véhicule et des points caractéristiques. Si un amer est observé plusieurs fois, son incertitude diminue. En revanche, si un amer apparaît pour la première fois (détection d'un nouveau point caractéristique) alors la carte globale est complétée et l'incertitude sur ce point est fixée par rapport à cette première mesure.

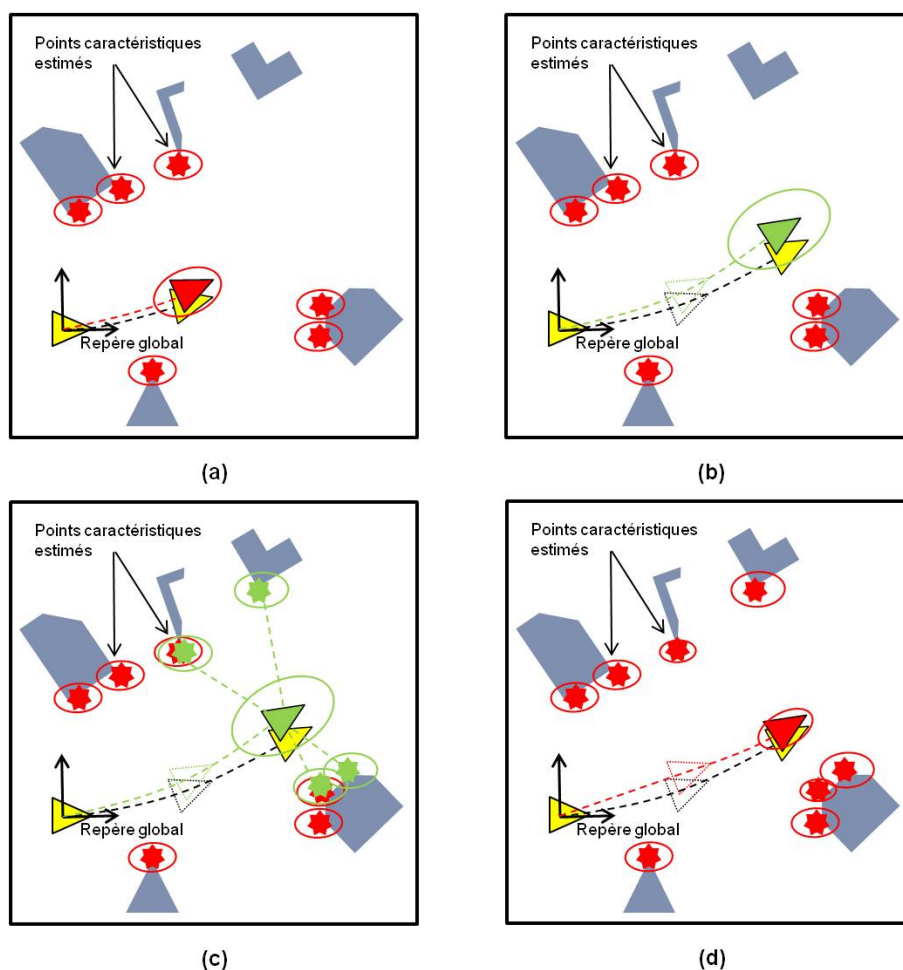


FIGURE 1.11 – Algorithme général du SLAM : (a) phase initiale, (b) phase prédictive, (c) phase d'acquisition et (d) phase d'estimation et de cartographie.

### 5.3 Cartographie

La construction de la carte joue un rôle important dans le SLAM car elle va contenir toutes les informations nécessaires à la localisation du robot et des amers existants et à venir. Ce sera également une étape primordiale pour la constitution d'un archivage cartographique.

Il est néanmoins important de comprendre que « cartographie » signifie, du point de vue de la navigation des véhicules autonomes, une carte qui décrit l'environnement en termes de types d'objets qui peuvent être détectés par les capteurs extéroceptifs.

Cela peut signifier qu'une carte de localisation peut, par exemple, être représentée par une liste de positions d'amers susceptibles d'être détectés lors d'une navigation. De plus, d'autres objets non détectés par les capteurs peuvent être présents dans l'environnement mais absents de la carte. Une carte de localisation ne liste donc pas forcément tous les obstacles existants (cf. figure 1.12).

La construction d'une carte exige de choisir un type de représentation de l'environnement. On peut les regrouper en quatre techniques :

- l'approche géométrique : utilise les primitives géométriques des amers ;

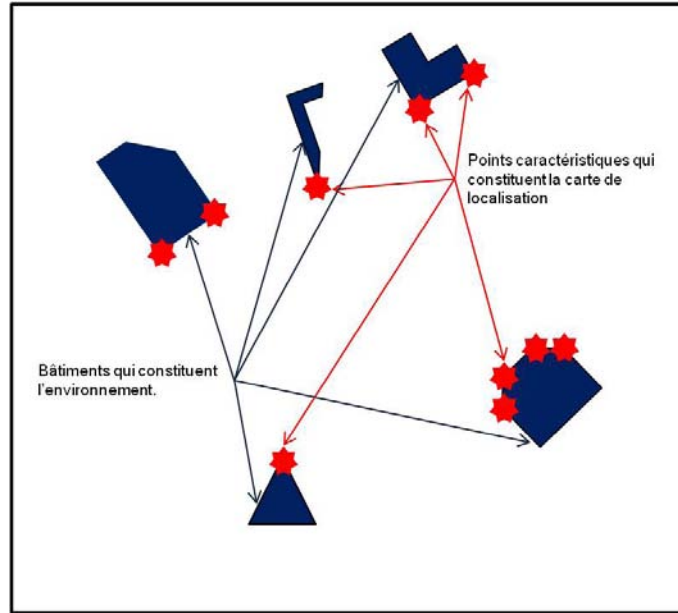


FIGURE 1.12 – Comparaison entre une carte de localisation et l’environnement réel. En bleu les éléments constituant l’environnement et en rouge les éléments de la carte.

- l’approche par grille d’occupation : représentation métrique qui décompose l’espace en un ensemble de cellules. A chacune des cellules est associée une probabilité d’occupation estimée à partir des données extéroceptives ;
- l’approche topologique : graphe qui contient toutes les informations sur la connexité et l’ordre des lieux visités par le véhicule ;
- l’approche directe : représentation de l’environnement par une utilisation directe des données acquises sans extraction de caractéristiques.

Ces quatre types de représentations peuvent être utilisés, chacun apportant des solutions à la résolution des différentes étapes du SLAM :

- choix du/des capteur(s) extéroceptifs ;
- extraction de données ;
- association de données ;
- gestion des incertitudes ;
- représentation de l’environnement ;
- gestion des boucles.

Après cette description des principes du SLAM, nous pouvons voir de quelle manière l’utilisation d’un radar à onde continue peut contribuer à l’amélioration du SLAM en environnement extérieur. Pour cela nous allons citer les contributions scientifiques de cette thèse.

## 6 Contributions scientifiques

Les principales contributions scientifiques de cette thèse s’intègrent dans une application qui est la mise en œuvre temps réel d’un radar *FMCW*<sup>25</sup> expérimental

<sup>25</sup>. Frequency Modulated Continuous Wave.



comme capteur extéroceptif d'un algorithme de SLAM.

Ce capteur autorise la détection d'objets situés dans un rayon de 100 mètres autour du véhicule. Il possède comme avantages par rapport à d'autres capteurs classiques (caméra, télémètre laser, etc.) de pouvoir être utilisé dans des conditions climatiques sévères (brouillard, pluie, obscurité, etc.) et de posséder une très bonne distance de détection. Son utilisation en SLAM est rarement abordée dans la littérature. On peut se reporter à l'article [Jose et Adams, 2004] qui en propose quelques exemples. Dans tous les cas, la méthodologie employée est du type *approche géométrique* avec une extraction des amers à partir des données radar.

Le travail décrit dans ce document s'attache à réaliser une étude du type *approche directe*, c'est-à-dire sans extraction d'amers et avec la prise en considération de toutes les informations contenues dans les données radar pour réaliser le SLAM. Cette approche, initiée par Lu et Milios [1994], existe depuis de nombreuses années mais n'a *a priori* pas encore été utilisée avec ce capteur.

Les contributions scientifiques de ce travail sont :

1. la création de panoramiques à partir de faisceaux radar. Ils fournissent en temps réel une représentation dense de l'environnement et vont s'intégrer dans des algorithmes de mise en correspondance afin d'alimenter un algorithme estimatif de localisation. Ils permettent de s'affranchir des difficultés liées à l'extraction d'informations dans les faisceaux radar, en utilisant une méthode directe par *scan matching*. Cela permet également d'offrir à terme, pour l'utilisateur, une carte de l'environnement aisément interprétable ;
2. la création d'un nouvel algorithme de mise en correspondance des panoramiques. Jamais utilisé en SLAM, cette méthode utilise les propriétés de la *Transformée de Fourier-Mellin* pour calculer la transformation rigide entre deux panoramiques. Cela indique les mouvements effectués par le véhicule entre les deux acquisitions. Cette information alimente l'algorithme estimatif de localisation ;
3. le calcul des incertitudes sur les mises en correspondance. C'est un aspect important souvent négligé. Il va permettre une meilleure adaptation de l'algorithme estimatif de localisation à la qualité des mises en correspondance ;
4. la mise en œuvre d'un algorithme estimatif de localisation qui utilise un filtre du type *EKF-SLAM* dans une *approche directe* de SLAM. Il est robuste aux problèmes d'association de données et de gestion des boucles ;
5. l'intégration du procédé global dans un applicatif temps réel multi-tâches qui exploite la bibliothèque d'optimisation *C++* qui se nomme *NT<sup>2</sup>*<sup>26</sup>. L'implémentation et les tests étant réalisés sur un véhicule d'exploration THALES.

## 7 Organisation du manuscrit

**Chapitre 2 :** il présente les principes du radar *FMCW* et son fonctionnement. Il indique également les techniques mises en œuvre pour le calcul des panoramiques qui seront utilisés comme données radars entrantes dans l'algorithme de SLAM.

---

26. <http://nt2.sourceforge.net/>.

**Chapitre 3 :** il donne un état de l’art du SLAM. Il présente les capteurs utilisés, les méthodes d’extraction des données et les algorithmes estimatifs les plus couramment utilisés dans la littérature SLAM. Il s’attarde plus précisément sur le capteur radar et sur le SLAM de type *approche directe*.

**Chapitre 4 :** il présente tous les procédés novateurs mis en œuvre. Cela débute par la description d’une nouvelle méthode de mise en correspondance des panoramiques radars. Cela se poursuit par la description de l’algorithme qui permet d’estimer la qualité des mises en correspondance (c’est-à-dire les incertitudes liées à la transformation rigide déduite). Cela se termine, par la présentation de l’algorithme estimatif de localisation, de type *EKF-SLAM*, appliqué à la mise en œuvre du radar, et par la description du calcul de la carte dense.

**Chapitre 5 :** il montre un ensemble de tests en situations réelles dans divers environnements vastes. Il présente ensuite une description des outils logiciels développés qui ont pour finalité l’intégration du procédé dans un véhicule d’exploration THALES afin d’aboutir à une mise en œuvre temps réel.

**Conclusion :** ce chapitre conclut ce travail et entame une discussion sur des perspectives possibles.



# Chapitre 2

## Les données du radar K2Pi

### 1 Introduction

Un véhicule autonome utilise pour sa navigation un ou plusieurs capteurs extéroceptifs qui vont fournir des informations sur l’environnement exploré.

Notre étude a pour objectif de montrer ce qu’apporte un radar comme capteur extéroceptif dans un procédé de localisation et cartographie simultanées. Le radar utilisé se nomme *radar K2Pi* [Rouveure et al., 2007]. Il a été entièrement conçu et réalisé par une équipe du Cemagref, partenaire du projet ANR IMPALA. Ce capteur est un radar à onde continue. Il fournit, toutes les secondes, aux algorithmes de localisation un panoramique, c’est-à-dire une description de l’environnement par une représentation numérique des informations réfléchies par les ondes radar. Ce format de données n’est pas traditionnel, habituellement le capteur et son unité de traitement fournissent des faisceaux que l’estimateur de trajectoire va seillonner pour en extraire des amers dans des directions données [Clark et Durrant-whyte, 1998 ; Dissanayake et al., 2001].

Il apparaît alors important de comprendre le fonctionnement d’un radar et d’expliquer comment à partir des ondes réfléchies par les objets ce dernier crée un panoramique, image de l’environnement.

Ce chapitre débute par une introduction sur le principe des radars, tournée plus particulièrement sur les radars à onde continue, utilisés dans la mesure de distance. Il donne une explication sur les influences de l’effet Doppler et du bruit de fond, dénommé *speckle*, sur les résultats. Il se termine par une description du format panoramique mis à disposition des algorithmes de localisation. Pour cela, il présente les problèmes liés à l’utilisation des capteurs tournants dans la robotique mobile et les solutions apportées dans le cadre du radar K2Pi.

### 2 Principe du radar *FMCW*

RADAR est l’acronyme de *RA*dio *DE*tectio*N* And *RA*nging qui signifie « Détection et Télémétrie Radioélectriques ». Un radar est un système qui utilise la propriété des ondes électromagnétiques de se réfléchir (en totalité ou partiellement) sur tout obstacle. Il permet ainsi de détecter des objets (cibles) qui sont situés à l’intérieur de son volume de couverture ceci afin d’en extraire des informations comme la position, la vitesse, et la forme.

Au sens large du terme, une cible est tout objet qui interfère avec l'onde émise et qui réfléchit une partie de l'énergie vers le radar. On fait la distinction entre une cible qui est l'écho d'un objet que l'on veut détecter et le « clutter » qui représente les échos non désirés (réflexions de la mer, de la terre, pluie, oiseaux, insectes, etc.) qui interceptent aussi l'énergie et la renvoient.

Une manière de mesurer la distance à un objet est d'émettre une courte impulsion de signal radio et de mesurer le temps que prend l'onde pour revenir après avoir été réfléchi. La distance est la moitié du temps de retour de l'onde (car le signal doit aller à la cible puis revenir) multipliée par la vitesse du signal (qui est proche de la vitesse de la lumière dans le vide si le milieu traversé est l'atmosphère) :

$$R = \frac{c\Delta t}{2} \quad (2.1)$$

avec  $c$  la célérité de l'onde dans le référentiel,  $R$  la distance mesurée et  $\Delta t$  le temps entre deux impulsions.

Pour la mesure de distances courtes (quelques dizaines de mètres), on remplace la technologie à onde pulsée, trop chère, par la technologie à onde continue [Monod, 1995 ; Noyman, 1996]. On mesure alors la distance du radar à la cible par l'utilisation d'une modulation de la fréquence. L'onde émise au temps  $T$  a une fréquence  $A$  mais une fréquence  $B$  au temps  $T'$  ultérieur. Le signal émis à  $T$  qui frappe une cible et revient au radar aura donc une fréquence différente de celle émise à ce moment là par le radar. En faisant la différence entre les deux fréquences, on peut déduire la distance parcourue, aller-retour, entre le radar et la cible.

Dans notre application nous utilisons un radar de type *FMCW* (*Frequency Modulated Continuous Wave*) qui permet de mesurer les distances à des obstacles. La fréquence du signal transmis change linéairement et forme une modulation triangulaire. La fréquence est modulée à un rythme  $f_m$  autour d'une fréquence  $f_0$  avec une variation totale  $\Delta f$ . Comme on peut le voir sur la figure 2.1 le signal reçu, représenté en pointillé (c'est-à-dire le signal réfléchi par une cible), est une réplique du signal transmis retardé d'un temps  $T$ .

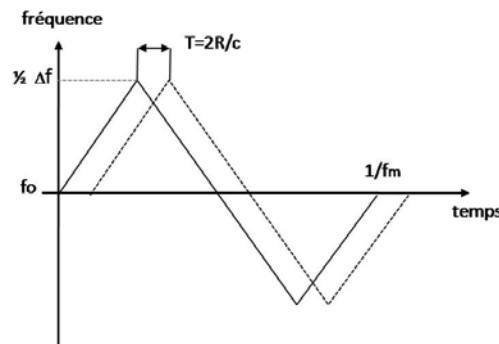


FIGURE 2.1 – Principe du radar FMCW pour la mesure des distances.

$$T = \frac{2 \times R}{c} \quad (2.2)$$

$$T = \frac{1}{\Delta f \times f_m} (f_r - f_e) \quad (2.3)$$

avec  $R$  la distance recherchée,  $c$  la célérité de l'onde dans le référentiel,  $f_e$  la fréquence émise et  $f_r$  la fréquence reçue.

On obtient finalement :

$$R = \frac{c}{2 \times \Delta f \times f_m} (f_r - f_e) \quad (2.4)$$

On appelle  $f_b = (f_r - f_e)$  la fréquence de battement. Si la cible n'est pas stationnaire, une fréquence de Doppler va venir se superposer à la fréquence de battement. Il faudra donc l'éliminer si on souhaite conserver des mesures de distance correctes.

## 2.1 Effet Doppler

L'effet Doppler est le décalage de fréquence d'une onde acoustique ou électromagnétique entre la mesure à l'émission et la mesure à la réception lorsque la distance entre l'émetteur et le récepteur varie au cours du temps.

On peut définir la modification de fréquence subie entre un objet émetteur (le radar) et un objet récepteur en mouvement (la cible) par :

$$f_{rec} = \frac{1 - \left(\frac{v_{rec}}{c}\right)}{1 - \left(\frac{v_{em}}{c}\right)} \cdot f_{em} \quad (2.5)$$

avec  $f_{em}$  la fréquence émise par l'objet émetteur,  $v_{em}$  la vitesse de l'objet émetteur,  $f_{rec}$  la fréquence reçue par l'objet récepteur,  $v_{rec}$  la vitesse de l'objet récepteur, et  $c$  la célérité de l'onde dans le référentiel.

Dans le cas du radar, si ce dernier est en mouvement ou si les cibles sont en mouvement, la valeur de la fréquence va varier en fonction de la vitesse. Cela va influencer le résultat de l'équation (2.4).

Le radar K2Pi ne dispose pas, comme certains radar *FMCW*, de la mesure de la vitesse relative aux cibles. Si on souhaite corriger la fréquence Doppler (équation (2.6)), il faut disposer d'un capteur proprioceptif qui indique la vitesse du véhicule. De cette façon, la correction des mouvements du radar par rapport à des cibles fixes est possible. En revanche, pour ce qui concerne les objets mobiles, aucune correction ne peut être faite. Ils sont considérés comme des bruits d'acquisition.

On peut approximer la fréquence Doppler par :

$$F_d = \frac{2 \times V}{\lambda} \quad (2.6)$$

avec  $V$  la vitesse radiale relative radar-cible,  $\lambda$  la longueur d'onde ( $\lambda = c/f_0$ ),  $c$  la célérité de l'onde dans le référentiel,  $f_0$  la fréquence centrale d'émission qui vaut 24 GHz pour le radar K2Pi, et  $F_d$  la valeur fréquentielle corrective à imputer au résultat final ( $f_r$ ).

## 2.2 Le speckle

Le speckle est un bruit particulier que l'on retrouve dans toutes les images radar. Il a pour origine une sommation d'un certain nombre d'échos ponctuels qui sont situés à des distances légèrement différentes les uns des autres par rapport au radar. La dispersion de ces distances est du même ordre que la longueur d'onde du signal,

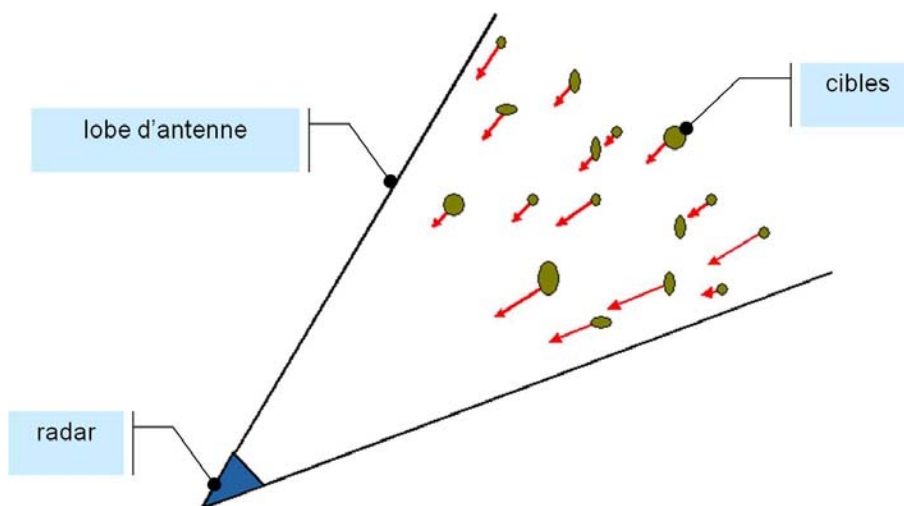


FIGURE 2.2 – Le signal radar mesuré correspond à la somme de toutes les fréquences renvoyées par les réflecteurs élémentaires présents dans le lobe de l'antenne.

la sommation des échos de la zone se fait de manière plus ou moins cohérente. D'un échantillon radar à l'autre on peut passer d'une sommation constructive (signaux en phase) à une sommation destructive (signaux en opposition de phase). D'un point de vue statistique, on obtient une sommation dont le résultat en terme d'amplitude répond à une loi de Rayleigh qui se traduit sur l'image par un aspect « sableux » que l'on appelle *speckle*.

Si on regarde la figure 2.2, on voit que l'antenne réceptrice du radar va recevoir des ondes réfléchies provenant de différents obstacles. La combinaison de ces diverses réflexions peut provoquer la création de fantômes (objets non présents mais créés par des réflexions d'antennes) ou la disparition d'objets (l'addition de signaux déphasés de  $180^\circ$  provoque une disparition de l'information). On retrouve cela sur la figure 2.3 où l'on se déplace le long d'un mur d'immeuble. On devrait retrouver dans le panoramique une droite représentant l'obstacle. En lieu et place, on retrouve des éléments discontinus. Cela est causé par l'addition de plusieurs signaux déphasés.

De façon générale, la littérature intègre également dans la terminologie *speckle* toutes les informations de bas niveau (données non informatives) présentes dans le panoramique.

### 3 Caractéristiques du radar K2Pi

Le radar K2Pi (cf. figure 2.4) a été conçu par l'équipe du CEMAGREF partenaire du projet IMPALA [Rouveure et al., 2006]. Il possède les caractéristiques décrites dans le tableau 2.1.

Il fournit 360 signaux radar (un par degré) dont il réalise l'acquisition par la rotation de son antenne à une vitesse radiale de 1 tr/s. Chaque signal acquis fournit la position des obstacles (dans un rayon de 100 m) qui réfléchissent l'onde radar. La figure 2.6 donne une représentation de ces signaux. Chaque faisceau est acquis pour une position angulaire donnée. Il est donc facile de passer à une représentation panoramique telle que le représente la figure 2.7. Cela correspond au passage d'une représentation polaire à une représentation cartésienne.

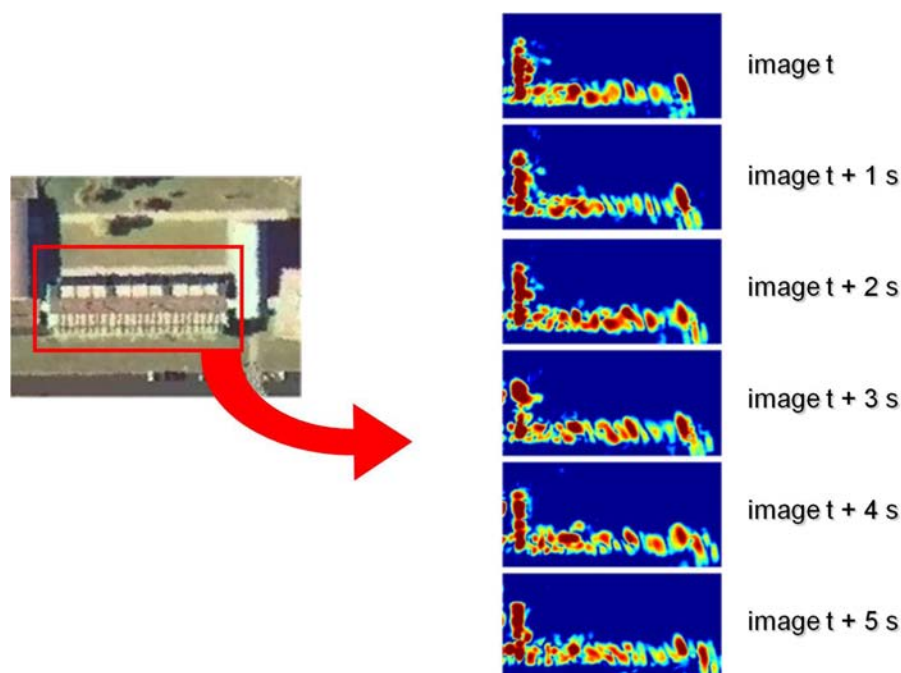


FIGURE 2.3 – Exemple d’acquisition radar sur une façade d’immeuble (données fournies par le constructeur du radar K2Pi). Les acquisitions aux instants successifs sont fortement bruitées.

TABLE 2.1 – Caractéristiques du radar FMCW K2Pi.

Porteuse $f_0$	24 GHz
Puissance d’émission $P_t$	20 dBm
Gain d’antenne $G$	20 dB
Bande passante	250 MHz
Résolution angulaire	3°
Précision angulaire	0.1°
Distance de détection Min/Max	3 m/100 m
Résolution en distance	0.6 m
Précision en distance	0.05 m
Dimensions (longueur-largeur-hauteur)	27-24-30 cm
Poids	10 kg

## 4 Calcul d’un panoramique corrigé

### 4.1 Utilisation des capteurs tournants dans la robotique mobile

Un capteur tournant effectue l’acquisition d’un faisceau pour chaque position angulaire (360 faisceaux, un par degré, acquis en une seconde par le radar K2Pi). Il est donc possible, si cette acquisition a été faite en statique, de représenter de façon précise tous les éléments détectés autour du capteur. En revanche, si on déplace le capteur lors de l’acquisition il apparaît difficile de repositionner chaque faisceau.

Afin de résoudre ce problème, on peut utiliser l’approche « stop & scan » [Nüchter et al., 2005 ; Ryde et Hu, 2007] qui consiste à stopper la plate-forme pour effectuer les acquisitions et de se déplacer ainsi, par à-coup. Ce type de déplacement n’est





FIGURE 2.4 – Radar K2Pi.

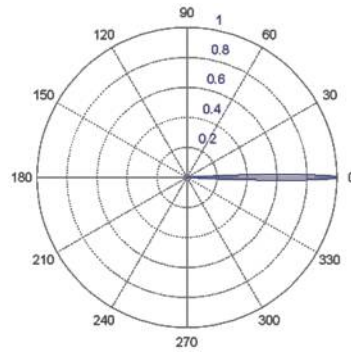


FIGURE 2.5 – Lobe d'antenne.



FIGURE 2.6 – Signaux radar : l'axe des abscisses représente la distance (0 à 100m) et l'axe des ordonnées l'angle de l'acquisition (0 à 360°).

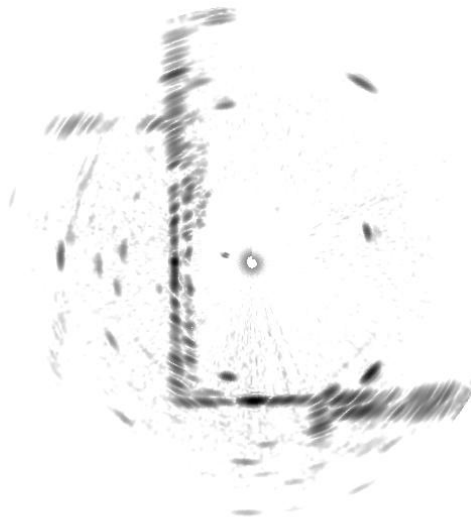


FIGURE 2.7 – Panoramique reconstruit (sans correction) à partir des signaux radar de la figure 2.6.

pas réaliste ni souhaitable pour un système pratique que l'on souhaite mobile. En effet, le fait de devoir stopper un véhicule pour chaque acquisition capteur limite grandement l'efficacité du système.

Une autre solution consiste à utiliser un capteur rotatif rapide afin de réduire l'effet de la distorsion et ainsi la négliger. Souvent prohibitif pour des raisons de coût, cette solution n'est pas envisageable sur le radar K2Pi qui est matériellement conçu pour une période de rotation d'une seconde.

Il est également possible d'utiliser des capteurs proprioceptifs qui vont fournir des informations sur le déplacement du capteur tournant lors de l'acquisition et donc permettre la correction de la distorsion. On suppose alors que les capteurs proprioceptifs sont parfaits. Cole et Newman [2006] ont choisi cette heuristique afin de rendre possible l'utilisation d'un laser 3D dans une application robotique réaliste. Pour corriger la distorsion, ils utilisent un odomètre et ne prennent en compte que les données acquises lorsqu'il n'y a pas de changement d'orientation. Ils considèrent en effet que les erreurs d'odométrie sont souvent liées aux changements de direction. Cependant si on souhaite travailler en toute connaissance de cause, les erreurs de ces capteurs doivent être prises en compte pour pouvoir générer une acquisition intègre.

Ainsi, classiquement, soit la distorsion est négligée dans le cas de capteurs rapides (ou de vitesses de déplacement lentes), soit le problème est contourné par l'utilisation d'une technique de « stop & scan », soit les panoramiques entiers sont recalés grâce à des données issues de capteurs supplémentaires.

Dans le cas du radar K2Pi, l'électronique et la mécanique du radar ne permettent pas d'accélérer les temps d'acquisition. De plus, une utilisation « stop & scan » est inconcevable dans un cadre réel d'utilisation. En définitive, afin de pouvoir, à partir des rayons acquis, construire un panoramique, on associe ce radar aux données de plusieurs capteurs proprioceptifs :

- un codeur angulaire donne l'angle de l'antenne lors de l'acquisition des rayons ;
- un odomètre indique la distance parcourue lors de l'acquisition du panoramique ( $d_i$ ) ;
- un gyromètre fournit le changement d'orientation entre deux panoramiques ( $\varphi_i$ ).

## 4.2 Construction d'un panoramique corrigé

Le radar fournit 360 faisceaux pour chaque acquisition. Il suffit à l'utilisateur d'associer ces rayons pour former un panoramique radar représentant une acquisition d'un rayon de 100 mètres centrée sur le véhicule.

Si on regarde la figure 2.8, cela correspond à interpoler les données radar (les faisceaux) qui sont dans le domaine polaire pour déterminer des valeurs dans le domaine cartésien du panoramique. Dans ce contexte statique, la figure 2.10 représente la position des valeurs cartésiennes (en bleu) à interpoler à partir des faisceaux radar (en rouge). Une interpolation bilinéaire peut-être utilisée pour déterminer les niveaux de chaque pixel du panoramique.

Si le véhicule est en mouvement (cf. figure 2.9), il faut modifier les lois d'interpolation afin de prendre en compte la variation des points origine des rayons au fur et à mesure de l'acquisition.

Il faut connaître avec exactitude la position du radar lors de l'acquisition de

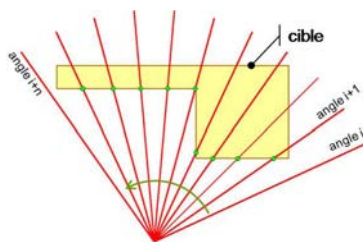


FIGURE 2.8 – Radar arrêté (données issues du simulateur radar du Cemagref).

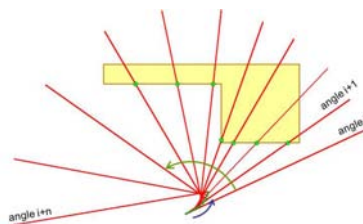


FIGURE 2.9 – Radar en mouvement (données issues du simulateur radar du Cemagref).

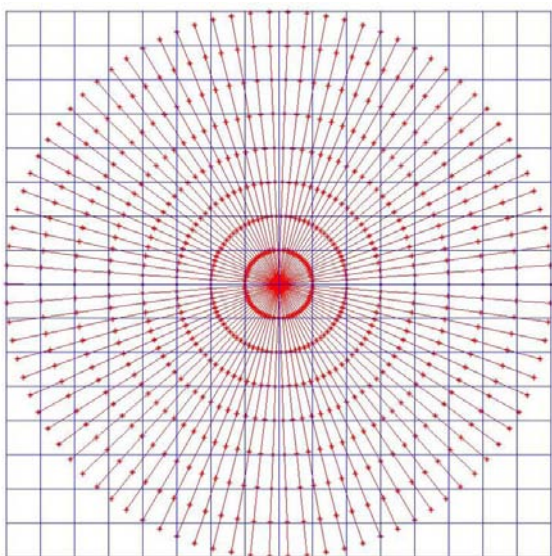


FIGURE 2.10 – Véhicule arrêté : exemple de 90 faisceaux de 8 valeurs (en rouge) à interpoler dans le domaine cartésien (en bleu) pour construire un panoramique.

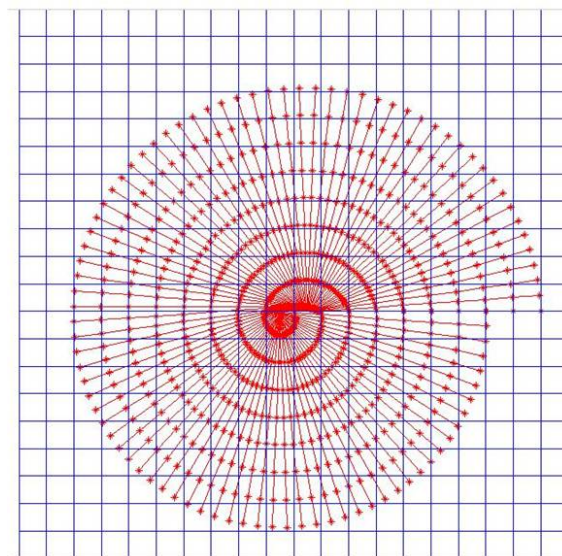


FIGURE 2.11 – Véhicule en mouvement : exemple de 90 faisceaux de 8 valeurs (en rouge) à interpoler dans le domaine cartésien (en bleu) pour construire un panoramique.

chacun des rayons (cf. figure 2.11).

Une approximation du déplacement du véhicule et l'utilisation de données proprioceptives (odomètre et gyromètre) permettent de fournir les positions successives du radar lors de l'acquisition des faisceaux (figure 2.12).

Sur la figure 2.12, le véhicule se déplace d'une distance  $d$  (mesure odométrique), il passe d'une orientation  $\varphi_i$  à une orientation  $\varphi_{i+1}$ . Il a subi un changement d'orientation  $\varphi = \varphi_i - \varphi_{i+1}$ . En utilisant un modèle de déplacement circulaire, on peut déterminer la position de chaque faisceau radar dans le repère de fin d'acquisition.

Si, par exemple, on cherche à déterminer la position du faisceau 120, alors on définit :

- $X_i$  : position du radar en début d'acquisition des rayons ;
- $X_{i+1}$  : position du radar en fin d'acquisition des 360 rayons ;
- $d$  : distance parcourue entre  $X_i$  et  $X_{i+1}$  ;
- $\varphi$  : variation d'angle entre la position  $X_i$  et la position  $X_{i+1}$  du radar ;

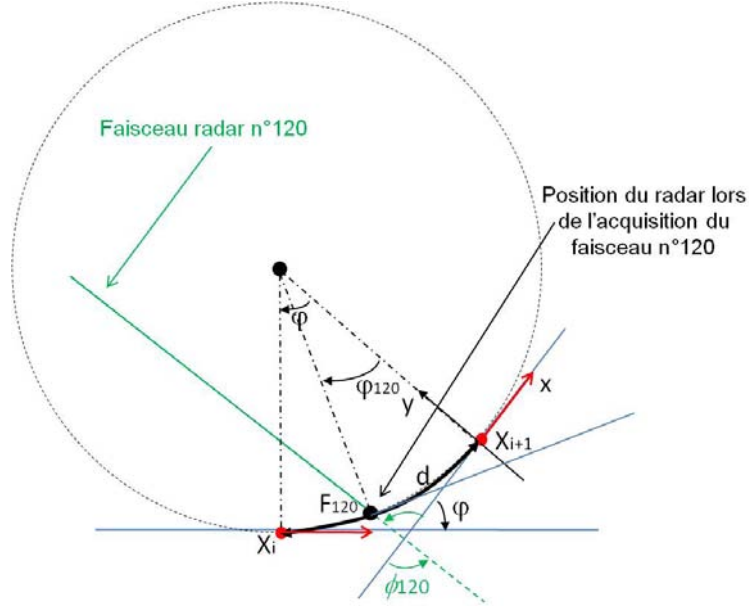


FIGURE 2.12 – Calcul de la position des faisceaux radar.

- $F_{120}$  : position du radar lors de la l'acquisition du rayon radar n°120 ;
- $\varphi_{120}$  : angle que fait le radar par rapport à sa position finale, lors de l'acquisition du rayon 120 ;
- $(x, y)$  : repère local positionné sur  $X_{i+1}$ .

On peut donc déduire la position et l'orientation du rayon n°120 dans le repère local positionné sur  $X_{i+1}$ , c'est-à-dire :  $(x_{120}, y_{120}, \phi_{120})$  :

$$Rayon = \frac{d}{\varphi} \quad (2.7)$$

$$\varphi_{120} = \varphi - 120 \times \frac{\varphi}{360} \quad (2.8)$$

$$x_{120} = -R \times \sin(\varphi_{120}) \quad (2.9)$$

$$y_{120} = -R \times (1 - \cos(\varphi_{120})) \quad (2.10)$$

$$\phi_{120} = \varphi_{120} + 2 \times \pi \times \frac{120}{360} \quad (2.11)$$

### 4.3 Correction de l'effet Doppler

Comme nous l'avons vu dans la section 2.1, l'effet Doppler introduit une erreur sur le calcul du positionnement des cibles lorsque le véhicule radar est en mouvement. Il faudra donc, à partir de la vitesse du véhicule (estimée par les informations de l'odomètre), calculer la fréquence Doppler puis corriger le faisceau correspondant.

Dans l'exemple de la figure 2.12, le faisceau 120 est soumis à l'effet Doppler engendré par le déplacement du véhicule. Si l'on nomme  $V_i$  et  $V_{i+1}$  la vitesse du véhicule respectivement aux instants  $i$  et  $i+1$ , on peut déterminer, par interpolation, la vitesse du véhicule lors de l'acquisition du faisceau 120 :

$$V_{F_{120}} = V_i + \frac{(V_{i+1} - V_i)}{t_{i+1} - t_i} \cdot \frac{120}{360} \quad (2.12)$$

La vitesse radiale a pour expression :

$$V_{rad_{F_{120}}} = V_{F_{120}} \times \cos\left(\frac{2 \times \pi \times 120}{360}\right) \quad (2.13)$$

A partir de cette vitesse on peut calculer la fréquence Doppler (voir l'équation 2.6) puis corriger le faisceau en conséquence.

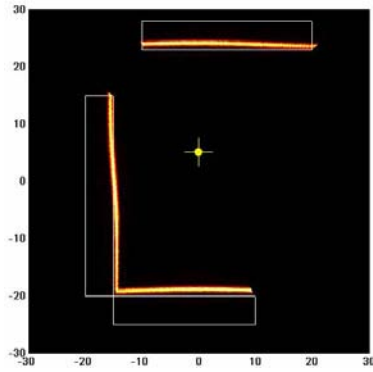


FIGURE 2.13 – sans correction Doppler (données issues du simulateur radar du Cemagref).

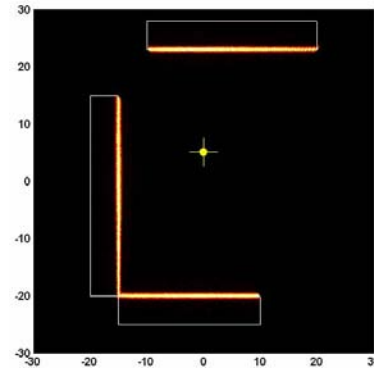


FIGURE 2.14 – avec correction Doppler (données issues du simulateur radar du Cemagref).

#### 4.4 Correction du cône de détection

Si on se réfère à la figure 2.2, on voit que l'information stockée correspond à la somme de toutes les réflexions produites à l'intérieur du lobe d'antenne. Dans ce lobe, la sensibilité du récepteur varie, on a une atténuation de 33dB sur les bords.

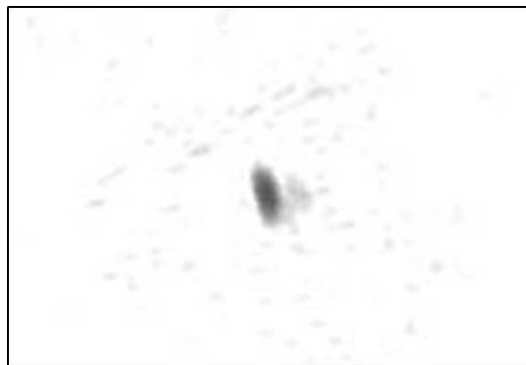


FIGURE 2.15 – Un poteau détecté par le radar. Les signaux sont mis sous la forme d'un panoramique, sans correction du cône de détection.

Si on réalise l'expérience de détecter un poteau, le panoramique reconstruit sans la correction du cône de détection va fournir une image semblable à celle de la figure 2.15.

Si on souhaite modéliser ce phénomène, on peut le simplifier en le représentant sous la forme d'un signal à une dimension (figure 2.16). Soit un signal réel  $f$ , un capteur qui possède une courbe de réponse  $H$  le détecte et fournit une image du signal

réel  $g$ . L'objectif est, connaissant  $g$ , de pouvoir via une méthode de déconvolution retrouver le signal originel  $f$ .

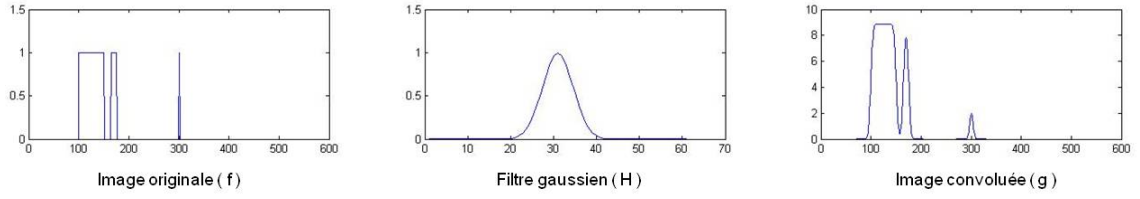


FIGURE 2.16 – Un signal  $f$  est convolué avec un filtre gaussien  $H$ , il en résulte un signal  $g$ .

Il existe plusieurs techniques pour réaliser la déconvolution d'un signal. Deux des méthodes sont le *filtre de Wiener* et l'algorithme de *Richardson-Lucy*.

### ***Filtre de Wiener***

Le filtre de Wiener caractérise le signal et le bruit par leurs propriétés statistiques. On considère que le signal est la réalisation d'un processus aléatoire stationnaire. On cherche alors à minimiser la moyenne du carré de la différence entre le signal et le signal restauré.

Si on considère un signal dégradé :  $f = h * u + b$  :

$u$  est le signal original à restaurer,

$h$  est un noyau de convolution symétrique positif (réponse impulsionnelle du capteur),

$b$  est un bruit gaussien d'écart type  $\sigma$ .

On modélise le problème par une formulation au sens des moindres carrés en considérant l'erreur quadratique  $\|f - h * u\|^2$  que l'on va minimiser. Le problème de minimisation n'admettant pas nécessairement de solution convenable, on ajoute un terme de régularisation quadratique de la forme  $\|q * u\|^2$ . Le noyau  $q$  sera fixé ultérieurement en fonction du rapport signal sur bruit.

### ***Algorithme de Richardson-Lucy***

L'algorithme de Richardson-Lucy est un algorithme itératif spatial. Comme précédemment, on considère un signal dégradé par un flou gaussien ( $h$  connu ou estimé) et du bruit  $b$  :  $f = h * u + b$ . On veut identifier  $u$ . L'équation de récurrence est :

$$u_{k+1} = u_k \left[ \frac{f}{(u_k * h)} * h^\vee \right] \quad (2.14)$$

avec par exemple  $u_0 = f$  et  $h^\vee(x, y) = h(-x, -y)$ .

La figure 2.17, montre la déconvolution du signal  $g$  de la figure 2.16 par une approche directe par filtre de *Wiener* et par la technique itérative *Lucy-Richardson*.

La figure 2.18 montre la forme du poteau 2.15 après déconvolution par l'algorithme de *Lucy-Richardson*.

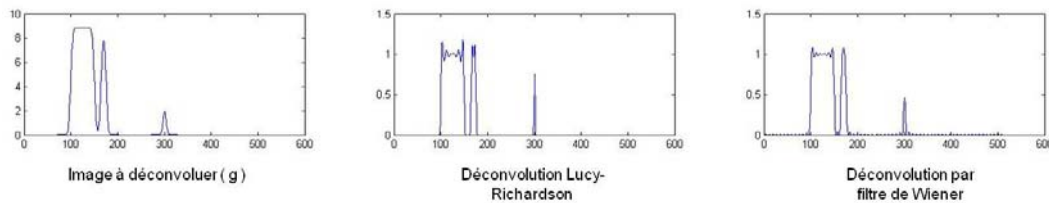


FIGURE 2.17 – Le signal  $g$  est déconvolué avec un filtre gaussien  $H$ , il en résulte un signal proche de la forme de  $f$ .

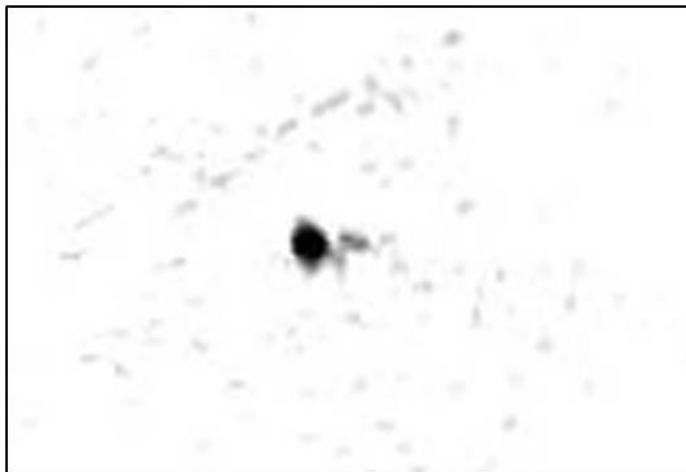


FIGURE 2.18 – Un poteau détecté par le radar avec correction du cône de détection par un algorithme de *Lucy-Richardson*

## 4.5 Correction du speckle

Le cône de détection de chaque faisceau va intégrer dans son résultat toutes les réflexions induites par les cibles, même les plus minimes. Cela provoque en résultat un effet de bruit, la création de fantômes, et la disparition d'informations. Dans ce contexte, Rouveure et al. [2006] nous proposent de moyennner  $N$  panoramiques successifs afin de minimiser cet effet (figure 2.19).

Dans un exemple concret (cf. figure 2.20), réalisé avec le radar K2Pi dans un environnement champêtre, on affiche deux panoramiques acquis à la même position. L'un a été reconstruit, sans correction du speckle, l'autre avec. Le résultat montre l'utilité de ce filtrage temporel. Le panoramique avec filtrage est représenté avec une forte minimisation des bruits et une correction des phénomènes de combinaison des fréquences réfléchies, notamment au niveau des bâtiments (phénomène présenté dans la figure 2.20). On voit néanmoins que les fortes saturations provoquées par des obstacles très réfléchissants sont amoindries mais ne disparaissent pas complètement.

## 4.6 Conclusion

Ce chapitre a permis de détailler les principes utilisés pour créer des panoramiques radar.

La mise en œuvre d'un capteur tournant sur un véhicule en mouvement entraîne certaines difficultés. Le robot change de position entre chaque acquisition de faisceau,



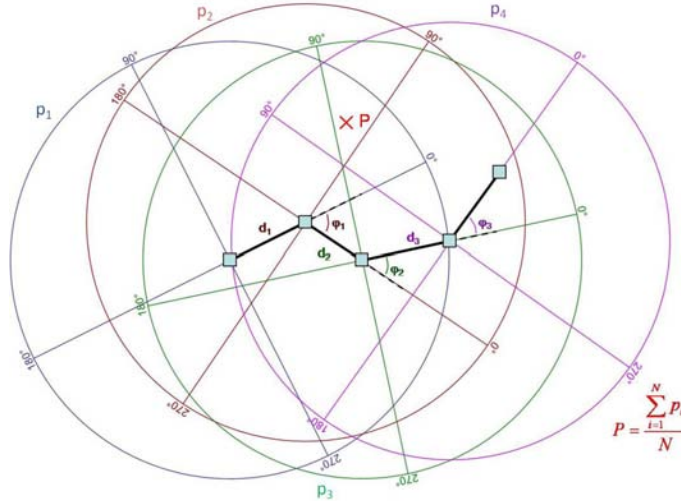


FIGURE 2.19 – On positionne  $p_1, p_2, p_3$ , dans le repère du panoramique  $p_4$ . On calcule  $P = \frac{\sum_{i=1}^4 p_i}{4}$  qui correspond au panoramique 4 après filtrage (données Cemagref).

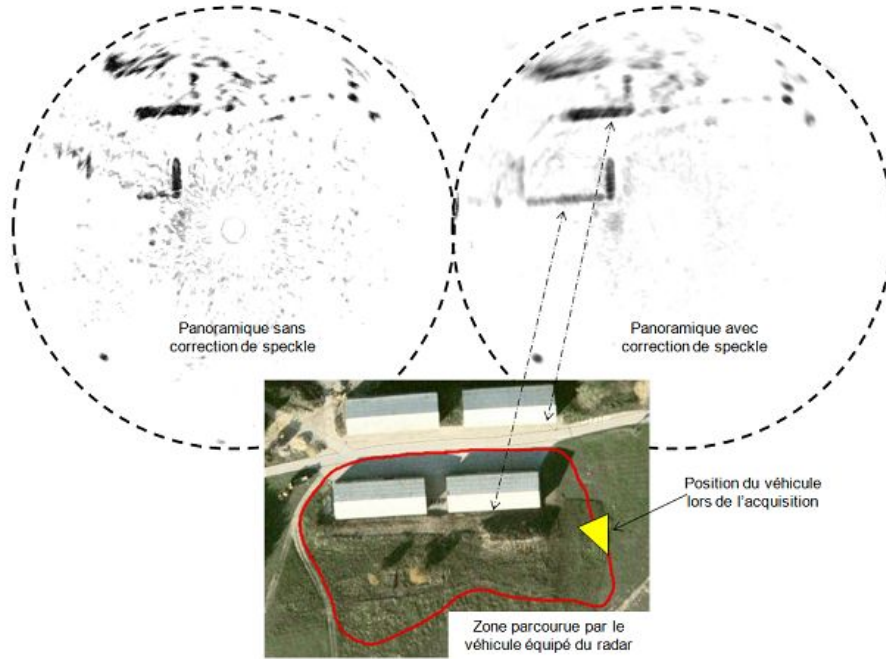


FIGURE 2.20 – Comparaison de panoramiques calculés avec et sans correction du speckle.

il faut donc prendre en compte cette information. Cela peut être intégré comme un paramètre de l'algorithme estimatif de localisation. Toutefois si l'on souhaite conserver l'aspect panoramique des données (qui est intéressant pour résoudre les problèmes de fort bruitage et d'association des données), cela est plus délicat.

Une des solutions pour connaître le parcours du véhicule est l'utilisation de capteurs proprioceptifs. De cette façon, il est possible de calculer la position du robot à chaque acquisition d'un faisceau. Cela permet de pouvoir calculer en temps réel les panoramiques. On suppose alors que les erreurs induites par l'odomètre et le



gyromètre sont faibles sur ces courtes distances.

On peut définir les étapes de la construction d'un panoramique :

- lecture des 360 faisceaux et des informations odométrique et gyrométrique ;
- estimation de la vitesse et du parcours du véhicule lors de l'acquisition ;
- correction du cône de détection ;
- correction de l'effet Doppler ;
- calcul du panoramique brut ;
- calcul du panoramique filtré par la sommation de  $N$  panoramiques bruts.

Le chapitre suivant s'attache à trouver dans la littérature SLAM des exemples pouvant s'appliquer au format de panoramique qui vient d'être décrit. C'est-à-dire des algorithmes de SLAM qui utilisent des panoramiques comme données extéroceptives. On s'attachera également à rechercher les différents cas d'utilisation d'une telle heuristique afin de voir les solutions apportées par les auteurs pour prendre en compte les erreurs des capteurs proprioceptifs utilisés.

# Chapitre 3

## SLAM - État de l'art

### 1 Introduction

Les recherches en localisation et cartographie simultanées (SLAM<sup>1</sup> ou CML<sup>2</sup>) datent de plus de deux décennies [Chatila et Laumond, 1985 ; Smith et al., 1988 ; Moutarlier et Chatila, 1989]. Il n'en demeure pas moins que les problèmes scientifiques liés à l'exploration de sites extérieurs étendus et l'association pertinente des données restent ouverts.

Dans ce contexte, l'utilisation d'un nouveau radar hyperfréquence FMCW fournissant des données de type panoramique constitue une nouveauté. Il ne s'agit plus d'extraire à partir de rayons des informations de type amers mais de trouver une technique qui va permettre de valoriser des informations intégrées dans des panoramiques. On peut se reporter au chapitre précédant pour se remémorer de façon précise les forces et faiblesses apportées par ce type de représentation. On peut citer le fait qu'un panoramique est calculé toutes les secondes et qu'il fournit une représentation exhaustive de l'environnement riche en information. Les problèmes viennent du fait que les informations sont plus difficiles à utiliser lorsqu'il s'agit d'extraire des données numériques. Un panoramique est issu de signaux acquis en mouvement et qui peuvent induire certaines erreurs. Un panoramique reste dépendant du fort bruitage des acquisitions radars notamment provoqué par des phénomènes de réflexion des signaux, de saturation, de speckle. Cela provoque l'apparition d'éléments parasites mais aussi une variation de taille et de forme des objets voire la disparition d'informations.

La littérature traitant de la mise en œuvre d'un radar FMCW qui fournit des panoramiques à un algorithme de localisation et cartographie simultanées est, à notre connaissance, inexistante.

On peut néanmoins trouver de nombreux travaux de recherche qui mettent en œuvre des capteurs extéroceptifs de type laser, caméra, sonar, et radar (hors cadre panoramique) dans des algorithmes de localisation et cartographie simultanées. Ils constituent une base de connaissance intéressante pour l'élaboration d'un SLAM qui utilise un radar FMCW fournissant des panoramiques.

La section 2 de cette étude présente un historique qui permet de comprendre la genèse du SLAM et de disposer rapidement de liens susceptibles de fournir de plus

---

1. Simultaneous Localisation And Mapping.

2. Concurrent Mapping and Localisation.

amples informations.

La littérature ayant pour thème le SLAM est très conséquente, de ce fait, la section 3 explique le choix effectué dans cette thèse au niveau de l'algorithme estimatif ainsi que les orientations choisies pour classer et limiter les investigations.

La section 4 présente les différents travaux de recherche qui peuvent apporter des solutions au problème posé. Enfin, la section 5 conclut ce chapitre en effectuant un constat sur l'existant et en proposant des solutions pour mener à bien les travaux de cette thèse.

## 2 Bref historique du SLAM

On peut se reporter aux documents de synthèse écrits par Durrant-Whyte et Bailey [Durrant-Whyte et Bailey, 2006 ; Bailey et Durrant-Whyte, 2006] pour avoir des explications sur les origines du SLAM et son fonctionnement.

La première formulation du SLAM probabiliste fut établie après la conférence « IEEE Robotics and Automation Conference » qui se déroula à San Francisco en 1986. Les travaux de Smith et Cheeseman [1986] et Durrant-Whyte [1988] établissent les bases statistiques de l'écriture mathématique reliant les amers<sup>3</sup> et les incertitudes de mesures. Ils montrent qu'il doit y avoir un fort degré de corrélation entre les positions estimées des amers observés. Ces corrélations doivent augmenter avec les observations successives. Dans le même temps, [Ayache et Faugeras, 1988] [Moutarlier et Chatila, 1989] et [Ayache et Faugeras, 1990] montrent des résultats de recherche sur la navigation de robots mobiles utilisant des algorithmes exploitant le filtre de Kalman. Il en résulte la parution d'un article clef du SLAM [Smith et al., 1990]. On y montre que lorsqu'un robot se déplace dans un environnement inconnu en effectuant des acquisitions successives d'observations d'amers, les estimations des positions des amers sont corrélées entre elles. En cause, l'incertitude sur la position du robot.

Les conclusions de ces avancées sont qu'un fort degré de corrélation entre les amers doit être maintenu si l'on souhaite conserver une cohérence dans les résultats. Une solution de SLAM consiste donc à utiliser un vecteur d'état contenant à la fois la position du robot et celles de tous les amers qui sont mis à jour après chaque observation. Elle demande une complexité de calcul qui croît très rapidement avec le nombre d'amers.

Une étape importante de l'évolution du SLAM est franchie avec [Csorba et al., 1996] qui fournit une étude théorique complète sur la convergence et la mise en œuvre du SLAM.

L'implémentation la plus connue représente les incertitudes comme une *fonction de densité de probabilité gaussienne* ( $PDF$ <sup>4</sup>) et propage ces incertitudes en utilisant un *filtre de Kalman étendu* ( $EKF$ <sup>5</sup>). Elle est connue sous le nom *EKF-SLAM* [Guivant et Nebot, 2001]. La popularité de cette approche est due au fait qu'elle est parfaitement adaptée au traitement de données extéroceptives de type amers. Elle fournit à la fois une solution récursive pour résoudre le problème de la navigation et une solution pour estimer, avec leurs incertitudes, les positions des amers et du

---

3. Cela se réfère à tous les points d'intérêt qui peuvent être observés par le robot.

4. Probability Density Function.

5. Extended Kalman Filter.

robot. De nombreuses solutions se fondent sur l'EKF-SLAM [Leonard et Durrant-whyte, 1991a ; Castellanos et al., 1999 ; Williams, 2001 ; Bailey, 2002 ; Guivant., 2002]. De nombreuses études s'attachent à améliorer, dans cette technique, les temps de calculs, la simplification des calculs matriciels mis en œuvre, et la consistance des résultats.

D'autres philosophies de traitements existent. Les premiers travaux introduits par Lu et Milios [1994] posent le problème comme une minimisation des erreurs sur les positions du véhicule. On associe à chacune des positions du véhicule une mesure extéroceptive. Cette méthode, proscrite à cause de temps de calculs trop longs, ne se fonde plus sur l'utilisation d'amers mais sur la mise en correspondance d'observations (nuages de points laser) permettant d'en retirer des informations de déplacement. On parle alors de SLAM orienté trajectoire. Elle est à l'origine du développement du SLAM hors-ligne qui permet de déterminer, en utilisant des techniques d'optimisation, la meilleure carte possible en connaissant toutes les observations et en utilisant les contraintes entre toutes les positions successives du robot. Thrun et al. [1998] utilisent notamment une approche bayésienne très efficace lors de la localisation avec une cartographie *a priori* mais qui ne permet pas de fournir un SLAM incrémental. Gutmann et Konolige [1999] présentent une des premières méthodes permettant de faire la mise à jour de la carte, à chaque nouvelle observation.

Il existe une troisième catégorie. Les techniques qui utilisent les filtres à particules. Elles reposent sur une factorisation suivant le théorème de Rao-Blackwell [Casella et Robert, 1996] où chaque particule représente une trajectoire possible et une carte. Montemerlo et al. [2002] utilisent cette méthode pour la construction d'une carte à base d'amers.

Pour terminer ce rapide historique, citons enfin quelques évolutions majeures :

- de nombreux travaux ont été développés pour limiter la complexité de la mise à jour de la carte, qui est en  $O(n^2)$ , avec  $n$  le nombre d'amers présents dans la carte, lorsque l'on utilise un filtre de Kalman. Citons les travaux [Guivant et Nebot, 2001] qui proposent une approche de type « batch processing » afin de mettre à jour la carte de manière différée sans que cela ait des conséquences sur la précision de la modélisation ;
- la notion de Hierarchical SLAM proposée par [Estrada et al., 2005] permet également d'alléger les calculs, en passant par des sous-cartes intégrées de manière hiérarchique dans une seule carte globale ;
- une carte stochastique est une représentation dite éparsée, car ne contenant que des amers utiles pour la localisation du robot (on peut utiliser le terme de carte de localisation). Il peut néanmoins être utile d'améliorer cette représentation afin de faciliter la planification des actions (par exemple si le choix d'une route s'effectue suite à l'interprétation de la carte par un opérateur). Dans ce cadre, de nombreuses recherches ont été menées pour améliorer la densité de représentation de la carte. C'est le cas des représentations par grilles d'occupation. On retrouve cela dans [Nieto et al., 2006] ;
- dans l'algorithme initial du SLAM, les amers sont supposés fixes. Que se passe-t-il si l'on prend en compte les objets mobiles au lieu de les ignorer ? On crée un algorithme de SLAM particulier, le SLAMMOT<sup>6</sup>, avec lequel un robot peut apprendre dans un environnement dynamique [Wang et Thorpe, 2002].

---

6. Simultaneous Localization, Mapping and Moving Object Tracking

### 3 Orientation de l'étude bibliographique

L'objectif de ces travaux de recherche est de tester un nouveau capteur extéroceptif de type radar panoramique hyperfréquence, et rentre dans le cadre d'expérimentations de SLAM en environnement extérieur.

Le but est de créer une carte pour le robot afin qu'il puisse se déplacer, se localiser, et réaliser des missions. Cette carte sera interprétée par l'homme afin de pouvoir diriger le robot de manière efficace. La notion de cartographie pour perception humaine prend donc tout son sens afin que les décisions de l'opérateur soient judicieuses.

Plusieurs orientations peuvent être choisies pour réaliser ces travaux. Afin de limiter le champ d'investigation on place plusieurs limites. Elles concernent l'algorithme de SLAM et les principes d'extraction de données utilisés.

#### 3.1 L'EKF-SLAM comme algorithme estimatif

L'objectif de ces travaux est d'expérimenter un nouveau capteur extéroceptif dans un SLAM en environnement extérieur. Dans ce contexte, on a volontairement choisi l'EKF-SLAM comme algorithme estimatif.

Il apparaît important de savoir si cela peut être préjudiciable au bon fonctionnement du procédé ou bien si cela peut lui être bénéfique.

Les caractéristiques de l'EKF-SLAM sont les suivantes :

- les distributions gaussiennes des variables sont représentées avec une moyenne et une matrice de covariance. Grâce à cette représentation, les mises en correspondance de ces variables avec les mesures recueillies sont réalisables facilement à condition que les estimations soient correctes ;
- son application est incrémentale. Une fusion des données extéroceptives et proprioceptives est réalisée à chaque itération ;
- la fermeture de la trajectoire du véhicule est implicite. La représentation des probabilités, via la matrice de covariance, permet de maintenir les corrélations entre les différents amers de la carte. Ce phénomène permet une convergence et une correction de toute la carte lors d'un retour sur une partie de l'environnement déjà cartographié ;
- les principales recherches menées sur ce type d'algorithme se sont faites en l'associant à des données de type amers ;
- dans l'approche classique, la complexité des calculs<sup>7</sup> est égale à  $O(n^2)$  pour chaque itération du filtre. Lorsque l'environnement visité est vaste, l'application en temps réel de l'approche est difficile ;
- le bruit sur les données est supposé gaussien. Dès lors que les mesures sont biaisées, les estimations obtenues par le filtre de Kalman divergent ;
- le filtre de Kalman est un filtre linéaire. Son application à des systèmes non-linéaires (comme dans un contexte SLAM) implique une linéarisation. Une telle utilisation provoque des problèmes de consistance des estimations et de l'incertitude sur celles-ci [Huang et Dissanayake, 2006].

---

7.  $n$  est le nombre d'amers présents dans la carte

En conclusion, l'EKF-SLAM convient à notre cadre d'utilisation puisqu'il permet une observation à chaque étape des estimées et une gestion des boucles. Il faudra néanmoins faire attention au problème des temps de calculs lors d'une utilisation sur de longs parcours.

Pour de plus amples informations, une description générale de l'EKF-SLAM est effectuée dans l'annexe A.

De plus, afin de s'assurer qu'il fonctionne aussi bien que d'autres types d'algorithmes, des tests comparatifs sont présentés au chapitre 5.

## 3.2 Le capteur extéroceptif dans la littérature SLAM

Le radar choisi fournit des panoramiques qui représentent directement son environnement lors de l'acquisition, ceci dans un rayon de 100 mètres. La figure 3.1 montre le cas d'un véhicule équipé d'un radar ayant effectué deux acquisitions successives. Ces représentations montrent l'aspect des informations fournies et permettent d'appréhender les méthodologies à mettre en œuvre.

L'état de l'art concernant le capteur extéroceptif et les méthodologies d'extraction associées ne va donc pas se limiter aux seuls cas d'utilisation d'un radar dans du SLAM. Il va aussi balayer un large panel d'applications afin de repérer des techniques pouvant être mises en œuvre pour le traitement de données extéroceptives de type panoramique.

Le plus simple est de débiter la recherche avec des capteurs de même nature (radar, sonar), puis de regarder des capteurs approchant les principes de détection (laser) : la détection d'un objet dans un axe donné se fait par réflexion d'une onde. Enfin, on peut voir les capteurs qui fournissent des informations proches d'une représentation panoramique, par exemple les caméras (utilisation dans le contexte des vues aériennes).



FIGURE 3.1 – Illustration de deux panoramiques successifs.

## 3.3 Taxinomie retenue

Le champ d'investigation est fixé, il reste à définir un principe de classement des nombreuses méthodes. En s'inspirant de la littérature, le choix est fait de classer le SLAM selon les méthodes de cartographie utilisées. En effet, on peut voir que toutes

les méthodes de SLAM étudiées ont en commun la création d'une cartographie, c'est-à-dire une représentation de l'environnement (du point de vue de la localisation). Cela fournit une référence pour les informations qui proviennent des capteurs extéroceptifs, garantissant après de multiples observations un positionnement avec une incertitude limitée.

Pour les méthodes jugées utilisables dans le contexte de cette thèse, une description des techniques d'extraction des informations issues des capteurs extéroceptifs est faite et une présentation des algorithmes issus du principe de l'EKF-SLAM est réalisée.

## 4 Etat de l'art

### 4.1 Introduction

L'état de l'art présenté dans cette section n'est pas exhaustif mais répertorie les principales solutions proposées dans la littérature. Ces solutions sont classées en fonction du type de représentation de l'environnement choisi.

Chaque méthodologie, si elle semble convenir à la résolution de notre problématique, sera décrite avec une attention toute particulière concernant :

- le mode de représentation de l'environnement,
- la gestion des incertitudes,
- la complexité algorithmique,
- le conditionnement des données extéroceptives, les caractéristiques des capteurs,
- la gestion des boucles.

### 4.2 SLAM avec une représentation par grille d'occupation

La grille d'occupation est une représentation de l'environnement dans lequel un robot mobile évolue, par décomposition de l'espace en un ensemble fini de cellules. La probabilité d'occupation de chaque cellule est une valeur estimée à partir des mesures fournies par les capteurs [Elfes, 1989 ; Schultz et Adams, 1998]. Un SLAM avec une représentation par grille d'occupation utilise un modèle capable de faire la mise à jour de l'environnement. Cela permet de réviser facilement les probabilités d'occupation et donc de suivre l'évolution de l'environnement autour du robot. De plus, la grille d'occupation est capable de modéliser des environnements de forme quelconque, et ne cherche pas à approximer les données par des primitives qui peuvent être inadéquates. Ainsi, elle fournit des informations d'occupation et les positions des obstacles. Elle est souvent utilisée lorsque l'application visée repose sur la connaissance de l'espace libre, en particulier pour la planification de trajectoires ou l'évitement d'obstacles. Elle est en général relativement facile à interpréter par l'homme.

La grille d'occupation constitue une représentation surfacique simple et populaire. Dans ce type de modèle, l'espace est discrétisé selon une grille régulière en cellules carrées ou rectangulaires de même taille. Chaque cellule contient une variable aléatoire, par exemple distribuée selon une loi gaussienne (cf. figure 3.2). Cette représentation peut être couplée à une estimation de probabilité maximale nommée

$EM^8$ . Cet algorithme statistique recherche la carte la plus probable dans l'espace de toutes les cartes possibles [Thrun et al., 1998]. L'utilisation de l'algorithme  $EM$  peut rendre l'approche coûteuse en temps d'exécution. Grisetti et al. [2005] proposent une méthode qui utilise un filtre à particules. Elle repose sur une factorisation suivant le théorème de Rao-Blackwell [Casella et Robert, 1996] où chaque particule représente une trajectoire possible et y associe une grille d'occupation. Le problème de cette technique réside dans la réduction du nombre de particules.



FIGURE 3.2 – Cartographie par grille d'occupation [Thrun, 2000].

Un des principaux obstacles de la grille d'occupation réside dans le fait qu'elle ne prend pas en compte la corrélation entre les cellules, corrélation causée par l'incertitude sur la position du robot. L'état de chaque cellule est indépendant. Dans les méthodologies précédemment présentées cela ne gêne pas puisque l'on ré-estime toutes les positions du robot à chaque étape, la carte étant ré-évaluée de la même façon. Dans le cas d'algorithmes tels que l'EKF-SLAM, la carte n'est pas représentée comme un tout, elle est constituée d'éléments autonomes (position des amers) reliés par des informations de corrélation.

En 1986, Elfes [1986] est le premier à utiliser cette cartographie, il l'associe à des capteurs ultrasons. On retrouve l'utilisation du sonar pour la construction de la grille d'occupation dans [Yamauchi, 1997]. Toutes ces cartes utilisent un modèle de capteur qui permet de calculer la probabilité d'occupation d'une cellule  $(x, y)$  à partir des mesures  $z$  ( $p(m_{x,y} | z)$ ). Cela représente le modèle inverse du capteur : il part des mesures des capteurs pour revenir à leur cause (la présence de l'objet). Tous ces travaux utilisant un sonar souffrent d'un même problème lié au caractère défaillant de ce capteur (fausses alarmes, détections manquées, difficulté de détection d'espaces étroits, données reflétées tardivement par une forme circulaire ou angulaire, ou même données diffusées face à une surface de verre, etc.). Pour résoudre cela, d'autres approches proposent de fusionner les données acquises par le sonar avec celles d'autres capteurs. Stepan et al. [2005] expérimentent la fusion « sonar-laser »

---

8. Expectation and Maximization est un algorithme composé de deux étapes : calcul des positions possibles du véhicule puis calcul de la carte la plus probable en fonction des prédictions précédentes.



et « sonar-caméra perspective » dans un environnement statique d'intérieur. Ceci afin de construire une grille d'occupation robuste permettant une meilleure planification. Ils concluent sur ces travaux en indiquant que le couple sonar-laser n'apporte rien alors que le laser utilisé seul fournit les mêmes résultats.

En définitive, la grille d'occupation est particulièrement bien adaptée à la représentation d'environnements encombrés, mais elle est moins efficace dans de grands espaces vides. De plus, la finesse de la discrétisation étant prédéfinie, elle n'est pas capable de s'adapter automatiquement à la densité ou à la taille des obstacles. En conséquence, si la grille d'occupation se prête bien à certains algorithmes de planification et d'évitement d'obstacles, elle peut cependant se révéler inefficace du fait du manque d'adaptation à l'échelle de l'environnement (en raison de la multiplication des cellules libres dans les espaces ouverts par exemple). On peut également reprocher à cette représentation un processus de positionnement complexe et coûteux notamment par rapport à une carte de balises où les objets à comparer sont moins nombreux.

#### 4.2.1 Conclusion

La grille d'occupation fournit une cartographie très dense. De plus, elle est très souvent utilisée avec des capteurs du type sonar car elle semble mieux adaptée aux caractéristiques de ce dernier : bruit important, mauvaise précision, fausses alarmes que d'autres méthodes telles que l'EKF-SLAM.

Elle semble donc pouvoir être implantée dans cette thèse qui utilise un radar comme capteur extéroceptif. Il faut néanmoins rappeler que l'algorithme estimatif choisi est l'EKF-SLAM. Son utilisation, associé à une grille d'occupation est impossible. En effet, dans une grille d'occupation la carte reflète un degré de probabilité de présence des objets sans aucun lien entre eux, alors qu'à l'opposé, dans l'EKF-SLAM, tous les objets sont liés entre eux. La grille d'occupation ne peut donc pas être choisie pour le SLAM radar à mettre en œuvre.

Néanmoins, si l'on reprend les premiers travaux sur la grille d'occupation qui concluaient que la cartographie par grille d'occupation doit résoudre le problème qui est de créer une carte la plus représentative possible de l'environnement à partir de mesures bruitées tout en connaissant les positions du robot, on peut alors considérer que la grille d'occupation est un outil idéal pour créer une représentation dense de l'environnement facilement interprétable par l'opérateur. Elle fournit une carte visuelle dans laquelle les cellules sont de la taille d'un pixel et les niveaux de gris représentent la probabilité de présence d'un objet dans l'environnement (cf. figure 3.3).

En conclusion, la représentation par grille d'occupation ne peut être utilisée dans le SLAM radar développé dans cette thèse. Elle est par contre parfaitement adaptée à la construction de la carte dense fournie à l'opérateur.

### 4.3 SLAM fondé sur une représentation topologique

Alors que la grille d'occupation est une carte métrique qui localise les éléments par des coordonnées cartésiennes, l'approche topologique représente l'environnement par des nœuds et des arcs qui correspondent aux trajets entre positions. Chaque nœud représente une zone d'intérêt. La carte indique simplement les liens (parcours)



FIGURE 3.3 – Carte dense orientée grille d'occupation.

entre les nœuds sans aucune information sur les positions absolues. La navigation entre deux zones non adjacentes est déterminée par un routage via les zones intermédiaires (cf. figure 3.4).

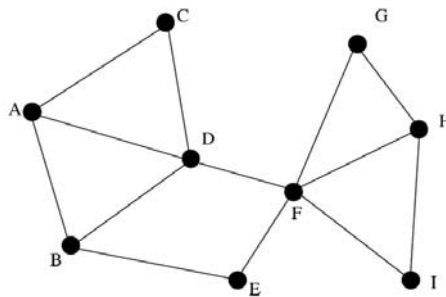


FIGURE 3.4 – Carte topologique [Bailey, 2002].

On peut distinguer deux méthodes pour la construire :

- on peut élaborer un modèle géométrique puis chercher ses propriétés topologiques, comme celui proposé par [Chatila et Laumond, 1985]. La définition du nœud est liée à une unité fonctionnelle ou topologique (corridor, bureau) alors que les connecteurs entre les nœuds représentent des portes ou des escaliers ;
- on peut également construire une carte topologique qui s'appuie sur des informations qualitatives ou physiques comme des couleurs ou des textures qui permettent de distinguer la nature physique d'une partie du terrain traversé, ceci afin de la considérer comme une entité discrète dans le graphe bâti. On peut rajouter éventuellement des informations métriques locales (distances et directions) sur des entités déjà perçues [Levitt et Lawton, 1990].

L'avantage principal des cartes topologiques est de faire abstraction des problèmes d'incertitudes dans le mouvement des robots : les incertitudes ne s'accumulent pas car le robot se contente de naviguer localement entre endroits. Ces cartes sont compactes, très efficaces pour la planification d'un cheminement dans la zone

d'exploration. Elles sont très utilisées en intérieur, lieu où l'on peut facilement découper l'environnement en zones d'intérêts (par exemple chaque pièce).

En revanche, l'absence d'informations géométriques peut empêcher le robot de réaliser certaines opérations. Comment choisir entre deux chemins possibles si on ne dispose pas d'informations de distances ? Comment faire si deux zones sont très ressemblantes pouvant ainsi générer des associations faussées ? Cela implique l'ajout de processus d'association de données très coûteux qui permettent de pallier cette difficulté.

Sur des lieux complexes, le découpage en zones s'avère délicat ou ambigu, l'absence d'informations métriques associées rend cette technique inutilisable telle quelle. C'est pour cette raison que les algorithmes qui utilisent cette méthode dans des environnements peu structurés la complètent avec des informations métriques [Kuipers et tai Byun, 1991 ; Thrun et Bücken, 1996 ; Jensfelt et Kristensen, 2001 ; Bailey, 2002].

La fermeture de boucle, problème récurrent du SLAM, ne permet pas ici à la carte de converger. Il faut associer à la carte topologique un processus annexe [Savelli et Kuipers, 2004 ; Beevers et Huang, 2005], ou bien rajouter des notions géométriques. On parle alors de méthodes hybrides. Elles représentent la carte par une méthode métrique couplée à une approche topologique [Tomatis et al., 2003 ; Kuipers et al., 2004]. Estrada et al. [2005] et Bosse et al. [2003] utilisent la carte topologique pour associer des cartes locales, cartes représentées avec une approche amers (voir la section 4.4).

#### 4.3.1 Conclusion

Une représentation par carte topologique n'utilise pas de notions géométriques mais permet un choix de cheminement facilité. Ne réalisant pas de gestion des incertitudes de positions, et n'apportant pas de réductions des erreurs en cas de bouclage, elle apparaît comme inadaptée au procédé étudié.

Dans le cadre d'une approche hybride, elle peut prétendre à être une étape dans le classement hiérarchique de cartes locales issues d'un EKF-SLAM.

### 4.4 SLAM avec une carte d'amers

L'environnement est représenté par des points d'intérêt [Leonard et Durrant-whyte, 1991b] que l'on nomme généralement « amers ». Ils ont comme principale caractéristique de pouvoir être détectés lors des observations via un capteur extéroceptif. Ils sont la représentation d'éléments existants sous forme de points, de lignes, de coins. Cette carte de localisation est une liste d'objets localisés. Ils sont décrits par des variables aléatoires suivant une loi de probabilité, généralement gaussienne. Ce type de représentation est appelé *carte stochastique* [Chatila et Laumond, 1985 ; Smith et al., 1988].

Smith et al. [1988] fournissent l'une des premières descriptions complètes de SLAM utilisant ce principe. Seules les positions des amers vis à vis du robot sont utilisées pour construire la carte (cf. figure 3.5). Néanmoins, des informations de forme, de couleur, peuvent être nécessaires pour résoudre les problèmes d'association de données.

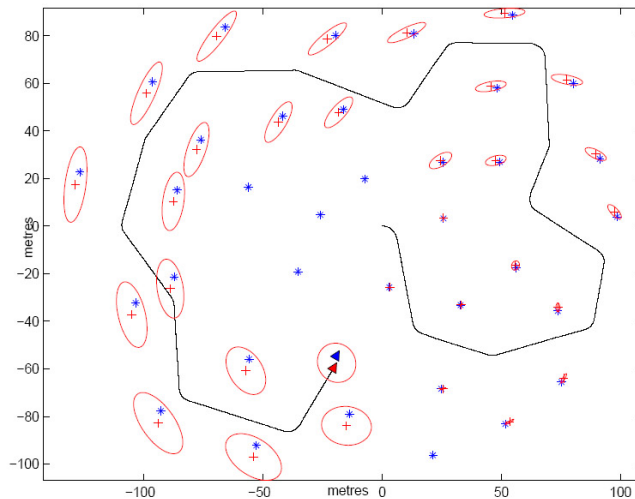


FIGURE 3.5 – Carte à base d'amers [Bailey et al., 2006] : les croix représentent les positions réelles des amers et les étoiles les positions estimées. Les ellipses indiquent le niveau d'incertitude associé à chaque estimation. Les triangles en bleu sont les positions réelles du robot et en rouge les positions estimées. En trait plein, on retrouve le parcours estimé.

On retrouve cette représentation par cartes d'amers dans de nombreuses implémentations temps réel du SLAM via l'EKF-SLAM [Leonard et Durrant-whyte, 1991a ; Bailey, 2002]. Cette méthode permet une convergence de toutes les estimations dès lors que le véhicule boucle sa trajectoire. Ceci est rendu possible par la présence des corrélations entre les éléments de la carte définies dans la matrice de variance-covariance [Castellanos et al., 1997].

En revanche, les cartes d'amers sont souvent peu denses et contrairement aux grilles d'occupation, elles ne contiennent pas les informations d'encombrement nécessaires pour la planification de trajectoires. En effet, les amers ponctuels renseignent peu sur la position des obstacles continus de grande dimension. Les amers de type segment ne sont généralement pas jointifs : le contour des obstacles est rarement continu et fermé, ce qui génère des ambiguïtés. Les cartes d'amers constituées par des amers géométriques structurés (segments, plans, etc.) voire par des objets sont plus faciles à comprendre que ceux avec des amers basiques (notamment des points).

Les points clef de ce mode de représentation se situent dans la qualité des algorithmes d'extraction et d'association des points d'intérêts.

Les environnements d'intérieurs ou urbains sont généralement plus adaptés à la détection d'amers.

#### 4.4.1 Les capteurs extéroceptifs

La navigation autonome en milieu extérieur reste à l'heure actuelle un domaine de recherche très actif, avec notamment les problèmes de détection et d'association des amers qui conduisent au bon ou au mauvais fonctionnement des algorithmes d'estimation. Une détection fiable des amers à partir de données extéroceptives bruitées doit permettre la convergence de l'algorithme et donc l'estimation des positions du véhicule et des amers.

Classiquement, on retrouve l'expression de la mesure selon le format  $(r, \theta)$  (cf. figure 3.6), fournie par un capteur de navigation tel que le sonar, le télémètre laser ou le radar. Ces données, associées à un bruit de mesure, correspondent à la position d'un amer supposé, relativement à la position du véhicule. Aucune autre information sur l'environnement n'est disponible. Ce signal peut donc correspondre à un amer ou à plusieurs amers, ou bien être simplement la manifestation du bruit environnant ce qui génère une fausse information. Ces ambiguïtés et l'association des amers entre acquisitions successives sont généralement levées par un algorithme d'association qui utilise un seuillage statistique fondé sur la prédiction de la position.

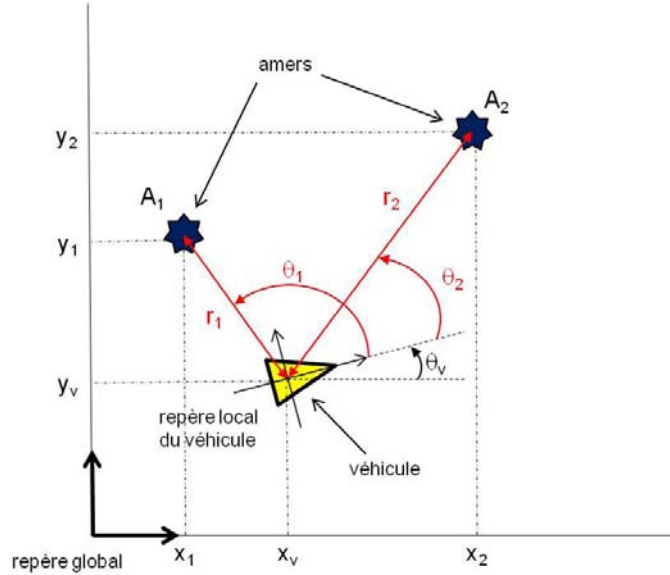


FIGURE 3.6 – Un véhicule positionné en  $(x_v, y_v, \theta_v)$ , observe un amer  $A_i$  et mesure la distance  $r_i$  qui les sépare et l'orientation  $\theta_i$  par rapport à son repère local.

Tardós et al. [2002] effectuent un comparatif des données laser qui sont denses et précises, et des données issues d'un sonar. Ils montrent la difficulté d'extraire des amers à partir d'informations sonar polluées (fausses détections, objets mobiles, mauvaise précision), et résolvent le problème d'association par la recherche de groupes d'informations via la transformée de Hough. Avec la même technique d'association, Ribas et al. [2007] expérimentent l'algorithme EKF-SLAM sur un sous-marin autonome équipé d'un sonar qui se déplace dans une zone portuaire (les données étant donc principalement constituées de lignes droites). Cette mise en œuvre dans un contexte réel assez favorable permet d'obtenir de bons résultats mais avec une gestion coûteuse des données.

Dans le cadre des radars, Clark et Durrant-whyte [1998] proposent un système de navigation qui utilise des réflecteurs artificiels comme amers. Dissanayake et al. [2001] combinent également les réflecteurs artificiels avec des amers naturels dans un SLAM utilisant un radar FMCW monté sur un véhicule. Clark [1999] présente une méthode de fusion de données issues de plusieurs véhicules en ne prenant en compte qu'une information par faisceau (celle dépassant un seuil minimum). Jose et Adams [2004] proposent une nouvelle méthode d'extraction des informations radar fondée sur une étude probabiliste. Ils montrent sa supériorité par rapport à des extractions à seuillage constant.

Le capteur doit disposer de plusieurs qualités : pouvoir discriminant, domaine de visibilité important, invariance et bonne adaptation à la métrologie. Dans la réalité, ces conditions ne sont pas toujours vraies et on se retrouve souvent avec un nombre important de primitives non idéales qu'il faut pouvoir discriminer afin d'effectuer les bonnes associations entre données. La méthode du plus proche voisin *NN* (*Nearest Neighbor* en anglais) trouve sa limite. Des algorithmes plus robustes sont nécessaires : Cox et Leonard [1994], et Smith [1998] utilisent des hypothèses de multi-trajectoires. Cela garantit une cohérence temporelle mais nécessite une carte par hypothèse. Bailey et al. [2000] utilisent des contraintes géométriques entre primitives pour obtenir de meilleures hypothèses. Neira et Tardós [2001] définissent un critère qui prend en compte de façon explicite les corrélations entre toutes les innovations et détermine les niveaux de compatibilités lors des appariements. Néanmoins, toutes ces méthodes, bien que réduisant les incertitudes, ne peuvent pas complètement supprimer les associations ambiguës.

La solution pour éliminer ces erreurs réside dans le fait qu'il est possible de fournir d'autres informations que la position d'un amer. On peut par exemple, définir un attribut de forme de l'amer : ainsi, si deux innovations indiquent un appariement mais que la forme des amers n'est pas la même, on peut rejeter l'association. Nieto et al. [2007] implémentent l'EKF-SLAM avec non plus des amers points mais avec des objets de formes multiples. Ils montrent la séparation entre le processus d'observation des données et celui d'estimation des positions futures via l'EKF-SLAM. La méthode d'association des formes, nommée *scan matching*, utilise notamment un algorithme de mise en correspondance de type *ICP*<sup>9</sup>.

Initialement les méthodes de *scan matching* étaient utilisées dans le cadre du SLAM visuel<sup>10</sup> pour la mise en correspondance de primitives. Argamon-engelson [1998], et Ulrich et Nourbakhsh [2000] utilisent la similarité entre images pour effectuer de la localisation sur carte. Newman et al. [2006] conjuguent *scan* laser et images pour la gestion des fermetures de boucles.

En marge de ces méthodes, on peut citer Mullane et al. [2009] qui utilisent un radar et proposent une méthode d'association des données intégrée dans l'algorithme d'estimation. Ils considèrent les informations capteur et les fausses détections comme des probabilités d'amers intégrés dans une grille d'occupation. Ce procédé fonctionne exclusivement en post-traitement.

#### 4.4.2 Algorithmes d'estimation

Le filtre EKF (Extended Kalman Filter) fut le premier estimateur présenté comme solution au problème du SLAM [Leonard et Durrant-whyte, 1991a]. La distribution gaussienne suivie par les variables du système à estimer, est représentée sous une forme continue. Une telle représentation facilite l'implémentation du filtre. En outre, elle permet une convergence de toutes les estimations dès lors que le véhicule boucle sa trajectoire. Ceci est rendu possible par la présence des corrélations entre les éléments de la carte définis dans la matrice de variance-covariance [Castellanos et al., 1997].

Très longtemps, les amers points furent les seuls à pouvoir être utilisés avec

9. Iterative Closest Point.

10. SLAM utilisant comme capteur extéroceptif une ou plusieurs caméras.

l'EKF-SLAM. Nieto et al. [2007] proposent une généralisation à des amers de formes diverses via le scan-SLAM. Ainsi, l'estimation de la position des amers est séparée de la forme de ces derniers. De ce fait, le scan-SLAM peut rapidement être applicable à d'autres algorithmes estimatifs, comme les filtres à particules [Burgard et al., 2003]. Bar-Shalom et al. [2002] proposent une application du filtre dual de Kalman nommé SEIF<sup>11</sup>, méthode améliorée par [Thrun et al., 2004]. Ce filtre est deux fois plus rapide que l'EKF et requiert quatre fois moins d'espace mémoire. Cependant, ses estimations sont moins précises que celles de l'EKF et souffrent d'inconsistance. L'ajout de l'estimation des positions du véhicule le long de sa trajectoire améliore la précision et la consistance des résultats mais augmente de manière considérable le temps de calcul [Eustice et al., 2006]. Cet ajout en fait une approche qui couple à la fois un SLAM orienté trajectoire (voir la section 4.5) du véhicule et une approche carte d'amers. On retrouve d'autres approches semblables telles que le *Graph SLAM* [Thrun et Montemerlo, 2006], *TreeMap* [Frese, 2006] et *SAM* [Dellaert et Kaess, 2006]. Ces méthodes fonctionnent bien dans les environnements extérieurs vastes mais elles nécessitent la connaissance globale des observations afin de pouvoir optimiser au mieux les calculs d'estimation des positions successives du robot.

#### 4.4.3 Conclusion

Les cartes d'amers géométriques sont peu adaptées pour représenter l'environnement extérieur ou alors seulement si celui-ci se limite à des régions pouvant être décrites sous forme d'éléments géométriques simples (points, segments, plans, etc.). Cependant, des approches utilisant la reconnaissance de formes des objets permettent d'améliorer le procédé.

Reste le problème essentiel du choix du capteur extéroceptif. Bien que quelques exemples existent, on s'aperçoit rapidement que le radar ne constitue pas un capteur de choix pour l'extraction d'amers. Permettant des détections par tous les temps et sur de longues distances, il reste trop sensible aux perturbations pour espérer une utilisation péri-urbaine sans ajout de réflecteurs artificiels. Si on se reporte au chapitre 2 sur le conditionnement des données radar, on retrouve des exemples illustrant cette difficulté qui est d'extraire d'un panoramique des amers.

La carte d'amers est la méthode de prédilection pour la mise en œuvre de l'EKF-SLAM, elle ne pourra pas être utilisée dans le contexte de cette thèse, la faute à des données proprioceptives qui sont inadéquates à l'extraction d'amers.

### 4.5 SLAM orienté trajectoire

Dans cette approche, le problème SLAM est présenté comme l'estimation des différentes positions du véhicule le long de sa trajectoire.

Une fois cette définition établie, on peut séparer ce mode de représentation en deux catégories distinctes :

- les méthodes issues d'une représentation à base d'amers. Le vecteur d'état contient les positions successives du robot et les positions des amers. Ces techniques hybrides ont été décrites dans la section 4.4. Elles se caractérisent par une utilisation courante des approches de résolution non-linéaires. On peut

---

11. *Sparse Extended Information Filter*.

citer : *Graph SLAM* [Thrun et Montemerlo, 2006], *TreeMap* [Frese, 2006] et *SAM* [Dellaert et Kaess, 2006], chacune essayant d'optimiser au mieux les calculs ;

- les méthodes qui s'inspirent de [Lu et Milios, 1994], dans lesquelles le vecteur d'état contient uniquement les positions successives du robot. On les retrouve sous les abréviations anglaises *pose graph SLAM*, *view-based SLAM*, *Direct SLAM*. L'article [Lu et Milios, 1994] montre la difficulté qui existe à réaliser l'association d'amers entre acquisitions successives. Pour résoudre cela, il représente la carte par une collection de poses robot chacune étant associée à un *scan*<sup>12</sup>. Pour déterminer les positions successives du robot, il emploie un système de minimisation non-linéaire qui utilise les contraintes existantes entre les *scans*. Deux *scans* proches représentent l'environnement du robot acquis à deux positions différentes. Une mise en correspondance des données permet d'en extraire le déplacement qui est effectué entre les deux acquisitions.

Cette section va s'attacher à étudier les méthodes issues de la deuxième catégorie, dans laquelle deux points novateurs distincts peuvent être cités :

- la technique de mise en correspondance, souvent citée sous le terme anglais *scan matching*, est un des points importants de la méthode. Elle va permettre, à partir des mesures extéroceptives (*scan*) acquises à chaque position du véhicule, de calculer les déplacements relatifs du véhicule entre deux poses. La qualité de la méthode de mise en correspondance est un facteur qui influe sur les résultats d'estimations finaux ;
- la technique de résolution du système. Elle va estimer les positions successives du véhicule à partir des différents déplacements qui sont calculés avec la technique de mise en correspondance. [Lu et Milios, 1994] et les méthodes associées utilisent plutôt une méthode de résolution non-linéaire (par exemple une minimisation par les moindres carrés), alors que d'autres préfèrent utiliser les matrices d'informations (inverse des matrices de covariance du filtre de kalman) avec des problèmes d'erreur de linéarisation ;

#### 4.5.1 Méthodes d'estimation

Konolige et al. [2010] indiquent qu'une *approche amers* va avoir tendance à surcharger le système en contraintes alors qu'une méthode avec des *scans* (cf. figure 3.7) va permettre d'éliminer du processus la partie extraction des données et va ainsi limiter le nombre de contraintes.

Le problème du SLAM peut être représenté par un graphique dans lequel les nœuds représentent les amers et/ou les positions du robot et les arcs les contraintes entre les nœuds. Finalement, que l'on utilise des amers ou des *scans*, seuls changent les liens entre les contraintes (cf. figure 3.8). Dans ce contexte, la littérature donne de nombreux exemples sur l'optimisation des algorithmes estimatifs. La plupart pouvant aussi bien s'appliquer sur du SLAM avec amers ou sur du SLAM avec *scans*.

12. Terme couramment utilisé pour définir un ensemble de points de mesure non dissociés. Cela peut-être tous les points d'impacts détectés par un laser SICK, une image vidéo, un panoramique sonar issu des niveaux des échos reçus, un panoramique radar constitué des niveaux des signaux réfléchis



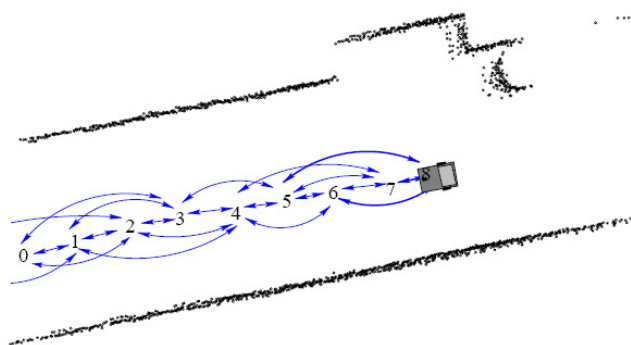


FIGURE 3.7 – Liens entre les poses successives [Gutmann et Konolige, 1999].

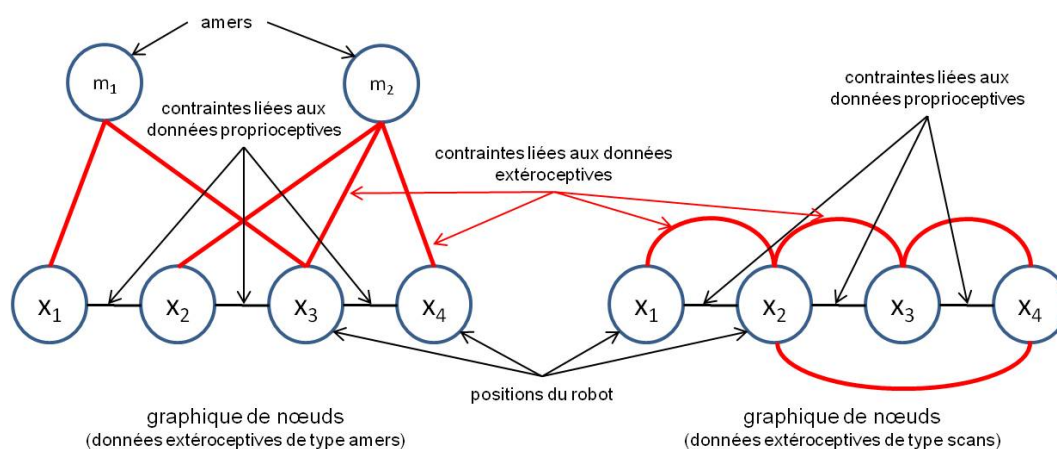


FIGURE 3.8 – Graphique de nœuds : représente de façon graphique les contraintes entre nœuds.

Depuis [Lu et Milios, 1994], de nombreuses approches permettant l'optimisation des graphes ont été proposées. La méthode la plus intuitive pour déterminer les positions successives d'un robot lorsque l'on connaît les contraintes (mise en correspondances des *scans*) les liant, est certainement une optimisation par les moindres carrés non-linéaires telle que celle de Levenberg-Marquardt. Cela demande de résoudre un système de taille importante (pour un SLAM de longue durée) coûteux en temps de calcul si on ne profite pas du fait que c'est un système creux. Konolige [2004] et Montemerlo et Thrun [2004] utilisent la technique *PCG*<sup>13</sup> pour optimiser la résolution du système de SLAM creux. Cette méthode nécessite  $n$  itérations pour converger, où  $n$  représente le nombre de variables. Frese et al. [2005] proposent une variante de la relaxation de Gauss-Seidel nommée MLR<sup>14</sup> qui résout les problèmes de lenteur de convergence. Olson et al. [2006] utilisent une technique de descente de gradient aléatoire<sup>15</sup> très facile à implémenter. Elle est très robuste dans le cas de mauvaises conditions initiales. Améliorée par [Grisetti et al., 2009], cette méthode souffre du fait qu'elle considère les erreurs de façon plus ou moins uniformes. Plus récemment, Dellaert et Kaess [2006] proposent une nouvelle méthode de résolution

13. Preconditioned Conjugate Gradient en anglais.

14. *Multi Level Relaxation* en anglais.

15. Plus connue sous le terme *stochastic gradient descent*.

de systèmes creux nommée  $\sqrt{SAM}$ . Kaess et al. [2007] fournissent une variante intitulée *iSAM* qui effectue une mise à jour incrémentale du système.

En contraste avec toutes ces méthodes d'optimisation non-linéaire, plusieurs chercheurs ont poursuivi l'étude des techniques de filtres. Eustice et al. [2006] proposent le *DSIF*<sup>16</sup> qui permet de résoudre, en temps réel, le système en utilisant une matrice d'information. Ceci grâce à l'introduction d'un faible nombre d'éléments à chaque itération. Cole et Newman [2006] montrent que l'utilisation du filtre de Kalman, malgré les problèmes de linéarisation et de résolution de matrices larges, reste une technique fiable pour réaliser un SLAM 3D dans un environnement extérieur. Ils utilisent comme données extéroceptives des scans laser qu'ils mettent en correspondance via un algorithme de type *ICP*.

Les observations utilisées sont de type laser [Newman et Ho, 2005 ; Cole et Newman, 2006], visuelles (données caméra) [Eustice et al., 2006], radar [Gérossier et al., 2009], et sonar [Burguera et al., 2010]. Elles nécessitent, pour tous les algorithmes précédemment cités, un *scan matching* efficace qui détermine les transformations rigides entre les observations. Elles indiquent quel mouvement  $[R + T]$  (Rotation + Translation) le véhicule effectue entre deux acquisitions. Ces informations sont ensuite utilisées comme contraintes dans le système à résoudre.

#### 4.5.2 *Scan matching*

Les algorithmes de SLAM qui permettent d'estimer au mieux la position d'un robot utilisent des observations pour mettre à jour l'estimation des positions. Dans le contexte d'une carte d'amers, les capteurs extéroceptifs fournissent la position relative des amers environnants dans le repère local du robot. Ils doivent également fournir une information sur les incertitudes des mesures effectuées.

Dans le cadre d'un algorithme de SLAM orienté trajectoire, les observations sont issues de la comparaison des *scans* acquis à chaque position du robot. L'incertitude sur les mesures, nécessaire à l'algorithme d'estimation, correspond donc à la précision de mise en correspondance des *scans* (cette valeur dépend de la qualité de détection du capteur et de la qualité de la mise en correspondance).

On peut représenter l'algorithme de *scan matching* via l'interface fonctionnelle suivante :

$$[\mathbf{RT}, \mathbf{Cov}] = \text{ScanMatch}(\mathbf{Scan}_{t_i}, \mathbf{Scan}_{t_{i-1}}) \quad (3.1)$$

$\mathbf{Scan}_{t_i}$  représente un *scan* effectué à l'instant  $t_i$ .

$\mathbf{RT}$  est la transformation rigide  $[R, T]$  à appliquer à  $\mathbf{Scan}_{t_i}$  pour le superposer avec  $\mathbf{Scan}_{t_{i-1}}$ .

$\mathbf{Cov}$  est la matrice de covariance associée à la transformation rigide déterminée à partir du procédé de minimisation. Elle représente l'incertitude en translation et en rotation sur l'exactitude du résultat fourni par la fonction *ScanMatch*.

On retrouve, dans cette modélisation, la notion de covariance. Elle apparaît indispensable pour intégrer le *scan matching* à un algorithme de SLAM du type EKF-SLAM dans lequel le processus de mise à jour (cf. chapitre 4 section 4.3) utilise une information d'innovation qui est fonction du  $RT_{reel}$  (fourni par le *scan*

<sup>16</sup>. Delayed Sparse Information Filter en anglais

*matching*) du  $RT_{estimate}$  (qui est déterminé par calcul), et des incertitudes sur les mesures. C'est l'intérêt de cet algorithme qui fournit une estimation de la position avec une valeur d'incertitude associée.

Plusieurs techniques sont présentées dans la littérature, elles vont être développées dans les paragraphes suivants.

### L'algorithme ICP

Le capteur le plus utilisé à l'heure actuelle en SLAM orienté trajectoire est l'imagerie laser qui délivre des nuages de points indiquant les distances à tous les objets proches. Il utilise généralement comme technique de mise en correspondance des méthodes issues des algorithmes « *Iterative Closest Point* » (ICP) [Besl et Mckay, 1992 ; Thrun et al., 2003 ; Cole et Newman, 2006] et « *Iterative Closest Line* » (ICL) [Bosse et al., 2003 ; Censi, 2008]. Ces algorithmes permettent de déterminer de façon itérative, généralement par minimisation de la distance euclidienne, la meilleure transformation possible pour mettre en correspondance des points entre deux acquisitions (cf. algorithme 1). Ils offrent l'avantage de pouvoir déterminer le couple translation-rotation composant une transformation rigide 2D (cf. figure 3.9). Ils demandent une valeur d'initialisation, généralement calculée à partir des données proprioceptives afin d'éviter de tomber dans des minima locaux.

---

#### Algorithme 1: Modèle général de l'ICP.

---

$[RT, Cov] = ScanMatch(Ref_{scan}, New_{scan}, RT_{init});$

$RT \leftarrow RT_{init};$

**tant que**  $k < Kmax$  *ou convergence non atteinte* **faire**

**pour** tous les points  $p \in New_{scan}$  **faire**

$p' \leftarrow TransformPOINT(p, RT);$

$q \leftarrow TrouverPlusProcheVoisin(Ref_{scan}, p');$

    Utiliser les points appariés  $[p, q]$  en entrée de l'algorithme de recalage en minimisant l'erreur quadratique ;

$$\epsilon = \frac{1}{N} \sum_{i=1}^N \|RT \cdot q_i - p_i\|^2;$$

    Il en résulte l'estimation  $\widehat{RT}$ ;

    mise à jour de  $RT = \widehat{RT}$ ;

$k = k + 1$ ;

Calculer la covariance  $Cov$

---

ICL [Besl et Mckay, 1992 ; Thrun et al., 2003] est similaire à ICP, excepté qu'il met en correspondance les points d'un *scan* avec des lignes extraites des points du *scan* initial. Cette approche est motivée par le fait qu'un LIDAR<sup>17</sup> lorsqu'il inspecte plusieurs fois un même environnement ne va pas forcément l'échantillonner de la même façon. En d'autres termes, les points entre deux *scans* peuvent différer mais les formes sont conservées. Cox [1991] présente une des premières études sur le sujet et donne des temps de traitements bien inférieurs à la seconde. C'est un des points important de ces méthodes.

L'estimation correcte de la covariance pour la technique ICP est indispensable si on

---

17. nom donnée aux imageurs laser, signifiant *Light Detection and Ranging* en anglais.

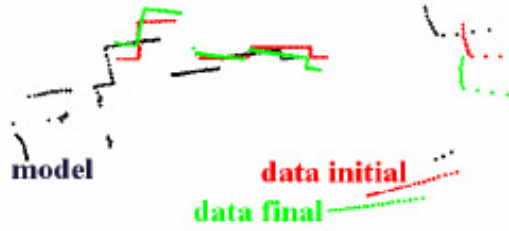


FIGURE 3.9 – Mise en correspondance de données laser. En noir les données de référence, en rouge les données courantes, en vert les données courantes après application de la transformée rigide calculée via ICP [Cole et Newman, 2006].

souhaite utiliser au mieux le *scan matching* comme mesure d'entrée d'un algorithme d'estimation. Toutes les méthodes existantes supposent que la mise en correspondance a fonctionné. Dans le cas inverse, par exemple si la minimisation récursive de l'ICP trouve un minimum incorrect (l'algorithme a abouti à un minimum local), les valeurs estimées de la covariance seront fausses et généralement optimistes.

On peut citer la *méthode directe* [Bengtsson et Baerfeldt, 2003 ; Censi, 2007a] qui peut-être décrite par l'algorithme 2.

---

**Algorithme 2:** Estimation de covariance par la méthode directe.

---

**début**

Réaliser l'acquisition du *scan*  $y_{t-1}$  effectué à la position robot  $s_{t-1}$ ;  
 Créer une carte  $\tilde{S}_{ref}$  à partir des données  $y_{t-1}$ , en connectant les points entre eux si la distance qui les sépare est inférieure à un seuil donné;  
**pour** un grand nombre de fois **faire**  
   Choisir un déplacement aléatoire  $\tilde{x}$  associé à une erreur de variance connue;  
   Simuler un nouveau *scan*  $\tilde{y}_t$  à partir de la connaissance de  $\tilde{S}_{ref}$  et à partir de la position  $s_t = s_{t-1} \oplus \tilde{x}$ ;  
   Mettre en œuvre l'algorithme de *scan matching* entre  $\tilde{y}_t$  et  $y_{t-1}$ .  
   Stocker l'erreur trouvée ( $\hat{x} - \tilde{x}$ );  
 Calculer la matrice de covariance à partir des valeurs expérimentales déterminées;

---

L'avantage de cette méthode est qu'elle ne dépend pas du procédé de mise en correspondance utilisé (reste la problématique de reconstruire une cartographie simulée correcte). Cependant, il apparaît impossible d'utiliser cet algorithme en temps réel à cause d'une quantité importante de calculs.

Dans la *méthode de transformation en amers*, Lu [1995] considère que chaque correspondance (chaque fois que l'algorithme ICP associe un point des données source avec un point des données destination) correspond à un amer observé. Il définit une

matrice de covariance  $P_{ij}^k = P_k^i + P_k^j$  (avec  $k$  le numéro de la correspondance traitée,  $i$  le point de l'image source et  $j$  le point de l'image destination) qui est calculée pour chaque correspondance comme la somme des covariances (des incertitudes de mesure sur ces points) des deux points exprimés dans le même repère. La matrice de covariance totale est estimée par :

$$cov(\hat{x}) \approx \left[ \sum_k (P_{ij}^k)^{-1} \right]^{-1} \quad (3.2)$$

Un des problèmes de cette méthode est qu'elle considère chaque point comme une observation indépendante alors que ce n'est pas le cas. Cela correspond à une estimation très optimiste.

Les *méthodes Hessiennes* telles que celle de [Bengtsson et Baerveldt, 2003] utilisent d'une part le fait que l'on peut linéariser la fonction d'erreur et d'autre part que la théorie sur les régressions linéaires peut être employée.

Soit une fonction  $Y = MX + w$  et son estimé  $\hat{Y} = M\hat{X}$ ,  $w$  est un bruit blanc de variance connue. L'erreur à minimiser est :

$$E(\hat{X}) = (Y - M\hat{X})^T (Y - M\hat{X}). \quad (3.3)$$

La théorie sur les régressions linéaires montre que cette fonction a une estimation optimale  $\hat{X}$  et une matrice de covariance associée qui a pour expression :

$$C(\hat{X}) = \left( \frac{1}{2}H \right)^{-1} \hat{\sigma}^2 \quad (3.4)$$

avec :

$$\hat{\sigma}^2 = \frac{E_{MIN}(X)}{n - k} \quad (3.5)$$

avec  $H$  la matrice hessienne,  $n$  le nombre d'observations et  $k$  le nombre de paramètres à estimer.

On peut appliquer ce résultat à l'algorithme ICP avec ses implications dans le *scan matching*. Soit :

- $P_i = (x_i, y_i)^T$  et  $P'_i = (x'_i, y'_i)^T$  les points mis en correspondance par ICP ;
- $(R, T)$  la transformation rigide estimée par ICP.

La minimisation de l'erreur revient à :

$$E(T, \theta) = \sum_{i=1}^N \left[ R(\theta)P_i + T - P'_i \right]^2. \quad (3.6)$$

Cette expression peut se linéariser dans le cas d'un angle de rotation petit. On se rapproche ainsi de la forme de l'équation (3.3).

Bosse et Zlot [2008] utilisent une approche similaire à celle décrite ci-dessus. Les différences se situent au niveau du calcul de la matrice Hessienne. Ils effectuent une linéarisation par un développement de Taylor du premier ordre autour d'une position initiale.

Ces méthodes peuvent dans certains cas sur-estimer la valeur de la covariance [Bengtsson et Baerveldt, 2003]. En effet, la forme de l'erreur pour une observation

donnée ne contient pas toutes les informations nécessaires à la détermination de la covariance vraie. Les équations précédentes donnent une fonction de minimisation qui dépend des paramètres estimés  $E_{MIN}(T, \theta)$  et non des erreurs sur les observations  $\tilde{z}$ .

Censi dans [Censi, 2007b] et [Censi, 2007a] présente une méthode d'estimation des covariances d'un algorithme de minimisation qui améliore l'estimation de la covariance en prenant en compte de façon explicite l'incertitude associée aux mesures  $cov(z)$ .

Soit  $\hat{x}$  le résultat d'un algorithme  $A$  minimisant une fonction d'erreur  $J$  qui dépend des observations bruitées  $\tilde{z}$  :  $\hat{x} = A(\tilde{z}) = \operatorname{argmin}_x J(\tilde{z}, \hat{x})$ . Alors la covariance de  $\hat{x}$  peut-être déterminée par une approximation du premier ordre [Censi, 2007a] :

$$Cov(\hat{x}) \approx \frac{\delta A}{\delta z} cov(z) \frac{\delta A^T}{\delta z} \quad (3.7)$$

ce qui donne :

$$Cov(\hat{x}) = \left( \frac{\delta J^2(\hat{x})}{\delta \hat{x}^2} \right)^{-1} \left( \frac{\delta J(\hat{x})}{\delta \hat{x} \delta z} \right) cov(z) \left( \frac{\delta J(\hat{x})}{\delta \hat{x} \delta z} \right)^T \left( \frac{\delta J^2(\hat{x})}{\delta \hat{x}^2} \right)^{-1} \quad (3.8)$$

Pour terminer, on peut citer une *méthode probabiliste* [Olson, 2009] qui propose une technique de *scan matching* temps réel formalisée selon une analyse probabiliste.

Le robot se déplace de la position  $x_{i-1}$  à la position  $x_i$  selon des informations proprioceptives  $u$ . Une observation  $z$  est fournie, elle dépend de l'environnement  $m$  et de la position du robot  $x_i$ .

L'objectif est de déterminer la distribution *a posteriori*  $p(x_i | x_{i-1}, u, m, z)$ . En utilisant les probabilités conditionnelles, [Olson, 2009] détermine la relation suivante :

$$p(x_i | x_{i-1}, u, m, z) \propto p(z | x_i, m) p(x_i | x_{i-1}, u) \quad (3.9)$$

$p(x_i | x_{i-1}, u)$  peut être déterminé en utilisant le modèle d'évolution du véhicule et les informations proprioceptives.

$p(z | x_i, m)$  est difficile à estimer car le modèle d'observation est fonction de la position actuelle et de l'environnement.

Olson [2009] calcule la distribution  $p(z | x_i, m)$  en utilisant des valeurs pré-calculées des probabilités du capteur utilisé, pour toutes les positions possibles. Il a donc réalisé une étude d'une multitude de cas typiques qu'il peut sélectionner suivant le contexte.

Une fois ceci réalisé, il est possible de déterminer la matrice de covariance sur les incertitudes du *scan matching*.

Étant donné  $x_i^{(j)}$  la  $j^{th}$  évaluation de  $x_i$ , si on fait l'hypothèse d'une distribution gaussienne, la covariance peut être estimée par :

$$\begin{aligned} cov(x_i) &= \frac{1}{s} K - \frac{1}{s^2} u u^T \\ K &= \sum x_i^{(j)} x_i^{(j)T} p(x_i^{(j)} | x_{i-1}, u, m, z)_j \\ u &= \sum x_i^{(j)} p(x_i^{(j)} | x_{i-1}, u, m, z)_j \\ s &= \sum p(x_i^{(j)} | x_{i-1}, u, m, z)_j \end{aligned} \quad (3.10)$$

En conclusion, les algorithmes qui utilisent l'ICP sont nombreux, ils permettent de déterminer la transformation rigide entre données issues de *scan* généralement laser. Ils fournissent également des techniques d'estimation de la matrice de covariance. Souvent pessimistes, elles offrent néanmoins l'avantage d'offrir au SLAM des valeurs d'incertitudes très utiles pour la consistance des calculs.

D'autres méthodes d'estimation de la transformation rigide existent. Elles sont moins utilisées et ne fournissent que très rarement une estimation de la covariance.

### Transformée de Hough

La transformée de Hough (*HT*) est utilisée dans le traitement d'image depuis les années 60. Elle permet de détecter, à partir de points, des formes de type droites ou cercles (cf. figure 3.10).

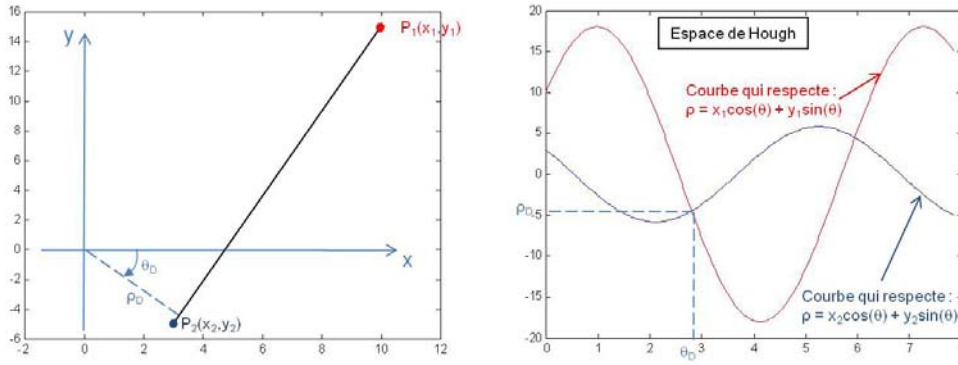


FIGURE 3.10 – Recherche des paramètres d'une droite passant par deux points.

Dans le cadre de la mise en correspondance de *scan*, il apparaît intéressant de voir comment réagit la *HT* à une transformation rigide de l'espace.

Ainsi, si on note  $I_{ref}(s)$  et  $I_{new}(s)$  deux entrées telles que  $I_{new}(s) = I_{ref}(R.s + T)$ , avec  $(R, T)$  la transformation rigide 2D.  $HT_{ref}(\theta, \rho)$  et  $HT_{new}(\theta, \rho)$  étant la représentation des données dans l'espace de Hough. On peut alors écrire :

$$HT_{new}(\theta, \rho) = HT_{ref}(\theta + \phi, \rho + (\cos \theta \sin \theta)T) \quad (3.11)$$

On peut remarquer (cf. figure 3.11) :

- si  $T = 0$ , alors  $HT_{new}(\theta, \rho) = HT_{ref}(\theta + \phi, \rho)$ , ce qui représente, dans l'espace de Hough, une simple translation dans la direction  $\theta$  ;
- si  $\phi = 0$ , alors dans l'espace de Hough cela se traduit par une incurvation dans la direction  $\rho$ .

On retrouve cette méthode dans [Censi et al., 2005], où il met en correspondance des *scans* laser. Il définit le spectre de Hough (*HS* : représentation spectrale de la transformée de Hough) afin de trouver une invariance sur la translation. Pour  $I_{new}(s) = I_{ref}(R.s + T)$ , il détermine :

$$HS_{new}(\theta) = HS_{ref}(\theta + \phi) \quad (3.12)$$

On peut en déduire l'algorithme de résolution suivant :

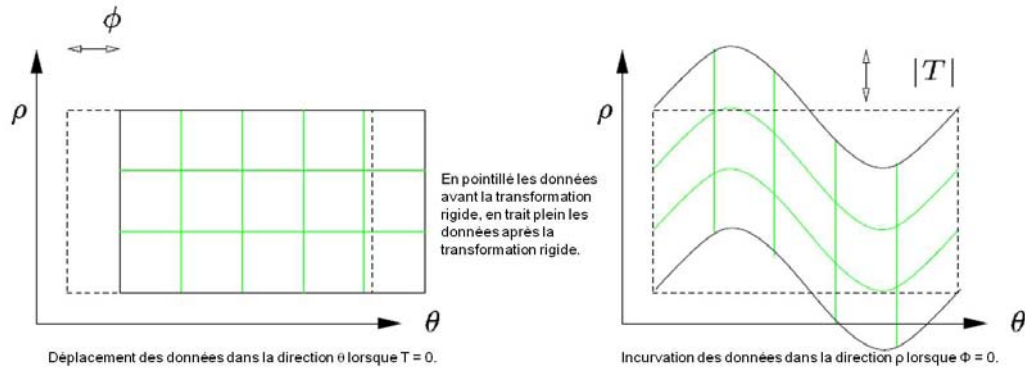


FIGURE 3.11 – Représentation dans l'espace de Hough des mouvements des données lors d'une transformation rigide  $R$  ou  $T$  [Censi et al., 2005].

---

**Algorithme 3:** Recherche de la transformation rigide par la méthode de [Censi et al., 2005].

---

**début**

- Recherche des transformées  $HT$  et  $HS$  des données  $I_{ref}$  et  $I_{new}$ ;
- Recherche des maxima locaux de la cross-corrélation pour déterminer les  $\phi$  les plus probables;
- pour chaque  $\phi$  faire**
  - Recherche de  $T$  par corrélation de la colonne de  $HT$  correspondante;
  - Détermination des solutions de  $T$  les plus vraisemblables

---

Il existe d'autres méthodes de Hough telles que [Ribas et al., 2007] qui l'utilise pour déterminer les transformations rigides issues de données sonar. Dans toutes ces techniques il faut toujours extraire du flux de données des informations caractéristiques. Plus délicat dans le contexte du sonar, car plus perturbé, cela s'effectue par la recherche des maxima locaux dans les faisceaux reçus.

En conclusion, cette méthode offre une nouvelle possibilité de détermination des transformations rigides entre *scans*. Fiable, elle souffre d'une part de sa dépendance à une extraction correcte des données reçues et d'autre part de sa difficulté à lever les ambiguïtés dans des environnements très denses.

### Recherche de la transformation rigide par histogramme

Cette approche [Wetzler et von Puttkamer, 1994] utilise l'idée générale selon laquelle le *scan matching* a pour objectif de trouver les caractéristiques qui restent inchangées dans plusieurs *scans* acquis dans le même environnement à des endroits et à des instants distincts. Ces caractéristiques forment une signature commune à tous les *scans*. D'où l'idée de créer des histogrammes qui regroupent pour chaque *scan*, les distributions des informations d'angle et de distance. La corrélation de deux histogrammes permettant de déterminer la transformations rigide ( $R, T$ ).

Röfer [2002] réalise le *scan matching* en deux temps. Il effectue dans un premier temps la résolution de l'orientation puis dans un deuxième temps la détermination de la translation. Pour se faire, il extrait des données laser des segments de droites (association de points voisins) auxquels il associe une orientation par rapport à l'axe



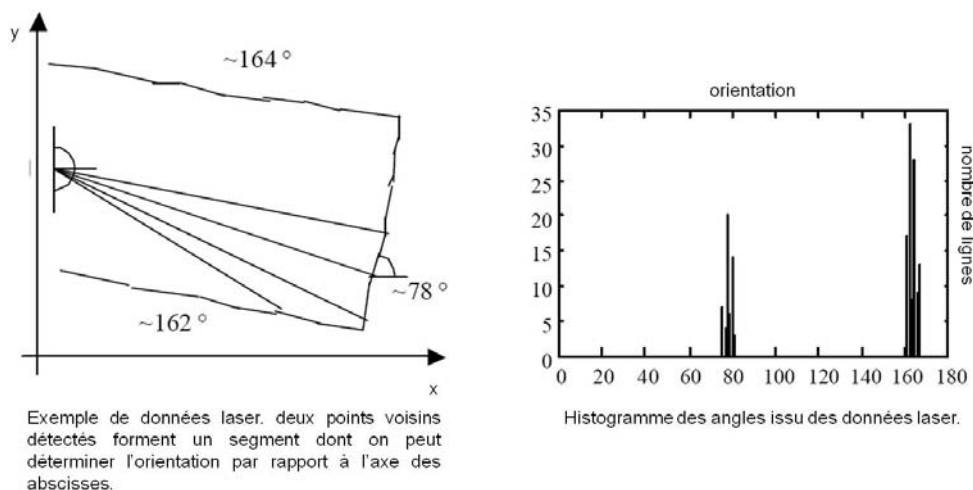


FIGURE 3.12 – Histogramme des angles [Röfer, 2002].

des abscisses. Il peut, de cette manière, calculer un histogramme des angles (cf. figure 3.12). De la même façon, il détermine un histogramme des distances à l'axe  $x$  et un histogramme des distances à l'axe  $y$ .

L'algorithme 4 donne les grandes étapes de résolution du procédé.

En conclusion, cette méthode fonctionne parfaitement pour un SLAM en intérieur, lorsque le robot évolue dans un environnement géométrique défini par des segments de droites. On peut cependant constater qu'il est, comme beaucoup d'autres algorithmes de *scan matching*, sensible aux ambiguïtés créées par des corridors, des pièces circulaires, ou tout autre environnement riche en symétries.

Pour terminer ce descriptif des méthodes de *scan matching* couramment utilisées en SLAM, on peut citer [Tomono, 2004] qui nécessite l'extraction de primitives au sein du *scan*, [Konolige et Chou, 1999] et [Olson, 2009] qui réalisent une corrélation par une technique markovienne (CBML<sup>18</sup>) entre le *scan* et la carte.

### 4.5.3 Conclusion

Les cartes orientées trajectoires avec *scan matching* sont bien adaptées aux données extéroceptives qui ne permettent que difficilement l'extraction d'amers. C'est le cas des panoramiques radar utilisés dans cette thèse. Néanmoins, la plupart des méthodes de mise en correspondance nécessitent un pré-traitement du type extraction de points ou de segments pour débiter l'algorithme. Cela implique deux choses : la capacité à différencier au maximum les données au sein du bruit et l'acceptation du fait que les niveaux de réflexion, mise à part pour le seuillage, n'apportent pas d'informations. Ils ne sont pas considérés comme une valeur ajoutée tel que peut le faire une image de caméra en niveaux de gris par rapport à une image binaire. En conclusion, aucune méthode ne semble réellement adaptée au cas de panoramiques issus d'un radar FMCW.

Enfin, bien que la littérature SLAM orienté trajectoire soit surtout constituée d'applications utilisant des algorithmes d'estimation fondés sur un système de minimisation non linéaire, on retrouve quelques cas dont [Cole et Newman, 2006] qui

18. *Correlation-Base Markov Localization* en anglais

---

**Algorithme 4:** Recherche de la transformation rigide par la méthode de [Röfer, 2002].

---

**Données :**  $I_{ref}$  et  $I_{new}$  les *scan* à mettre en correspondance

**Résultat :**  $(R, T)$

**début**

- Placement de  $I_{ref}$  et  $I_{new}$  dans le même repère;
- Corrélation des histogrammes des angles de  $I_{ref}$  et  $I_{new}$  (cf. figure 3.13a);
- Recherche du maximum et détermination de l'angle le plus probable  $\delta\theta$ ;
- Rotation de  $I_{new}$  d'un angle  $\delta\theta$  (cf. figure 3.13b);
- Recherche de la direction principale des deux *scans*;
- Alignement des deux *scans* parallèlement à l'axe des abscisses et dans la direction principale (cf. figure 3.13c);
- Création des histogrammes en  $y$ ;
- Corrélation des histogrammes en  $y$  et recherche du maximum  $\delta y$ ;
- Décalage de  $y$  d'une distance  $\delta y$  (cf. figure 3.13d);
- Alignement des deux *scans* parallèlement à l'axe des abscisses et dans la direction secondaire (cf. figure 3.13e);
- Création des histogrammes en  $y$ ;
- Corrélation des histogrammes en  $y$  et recherche du maximum  $\delta x$ ;
- Détermination de la transformation rigide  $(\delta\theta, [\delta x, \delta y])$  (cf. figure 3.13f);

---

laissent apparaître la faisabilité de mettre en œuvre un filtre EKF-SLAM associé à un *scan matching*.

## 5 Conclusion

Notre application met en œuvre un radar hyperfréquence FMCW qui fournit des données de type panoramique à l'entrée d'un algorithme estimatif de type EKF-SLAM.

L'étude bibliographique menée dans la littérature SLAM apporte plusieurs éléments de réflexion pour orienter le travail.

On peut différencier dans le SLAM deux parties distinctes. Il y a d'une part le procédé qui va fournir les mesures extéroceptives et d'autre part l'algorithme estimatif. Cela signifie que l'on peut adapter un même algorithme aussi bien à des données de type amer qu'à des données directes issues d'un *scan matching*.

Concernant les capteurs couramment utilisés, on retrouve les imageurs laser, les caméras vidéos, les sonars et les radars. L'étude des différentes publications montre que la littérature ne fournit que peu de cas utilisant le radar. Ils sont tous associés à des méthodes d'extraction d'amers allant même jusqu'à l'utilisation de réflecteurs artificiels pour faciliter la détection [Clark et Durrant-whyte, 1998 ; Dissanayake et al., 2001]. Mullane et al. [2009] utilisent une extraction des amers par la recherche de maxima locaux dans les faisceaux radar. Ils associent cela à un algorithme qui

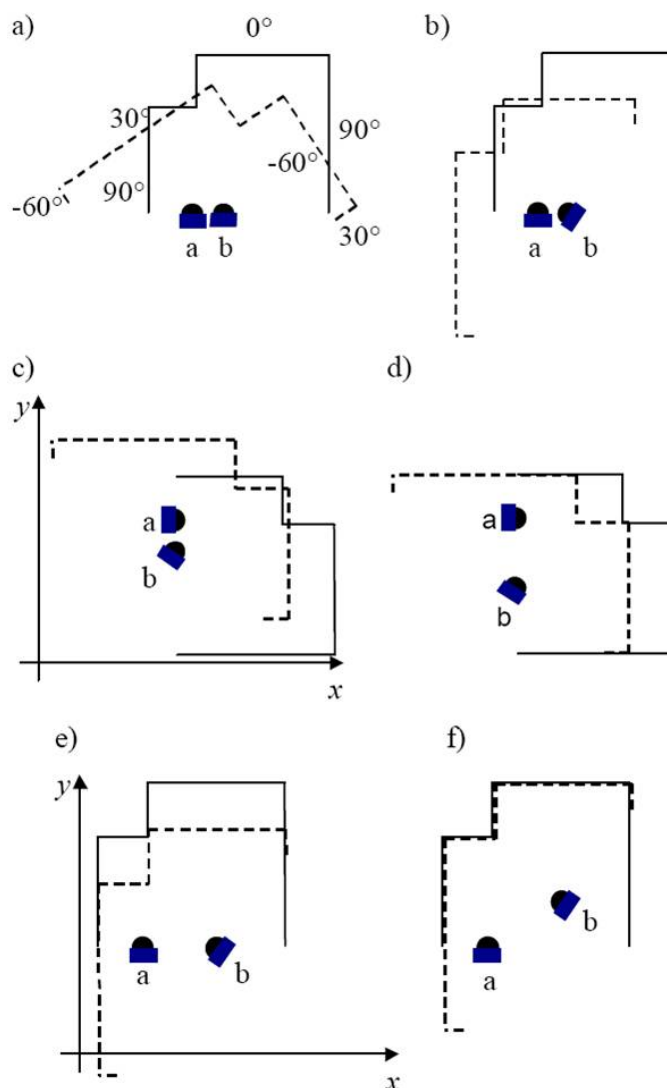


FIGURE 3.13 – Détermination de la transformation rigide par la méthode des histogrammes (cf. algorithme 4).

prend en compte les notions d'imprécision et de fausses détections intégrées à ce mode d'extraction. Une telle approche ne peut s'appliquer à un algorithme EKF-SLAM.

Si on s'attache à chercher des exemples d'utilisation avec le sonar qui reste, au niveau des signaux perçus, proche du radar (même si un sonar fournit des données moins précises et plus parasitées qu'un radar) on retrouve des exemples qui utilisent des grilles d'occupation intégrant directement les probabilités sur la validité des informations fournies [Elfes, 1986 ; Yamauchi, 1997]. Ces méthodes sont inutilisables associées à un EKF-SLAM. Tardós et al. [2002] et Ribas et al. [2007] réalisent l'extraction d'amers et la gestion des données par l'utilisation de la transformée de Hough, par contre ils testent leurs algorithmes soit en environnement intérieur soit dans un cadre favorable (submersible navigant dans des canaux rectilignes). Burguera et al. [2010] fournissent l'un des rares articles utilisant un sonar qui illustre une méthode directe par un *scan matching* de type ICP. Ils réalisent l'extraction des points fournis avec ICP par un seuillage intelligent qui s'adapte aux caractéristiques

de l'histogramme des données sonar. Cette méthode est mise en pratique par un submersible qui navigue dans des canaux portuaires rectilignes.

Enfin, si on se rapporte aux données des imageurs laser, on retombe sur un nombre abyssal de références illustrant le bien fondé d'utiliser l'ICP ainsi qu'une multitude d'algorithmes estimatifs optimisés.

En conclusion, si on souhaite extraire de ce flot de littérature une méthode qui peut s'appliquer sur des panoramiques radar, on voit que les méthodes d'extraction d'amers (dans des faisceaux radar) sont difficilement applicables dans un environnement péri-urbain (des données très denses, des informations parasites nombreuses, des formes d'obstacles variées) au vu des résultats fournis depuis ces vingt dernières années sur le sujet. Seule une méthode globale utilisant tout le panoramique dans une approche *scan matching* convient et plus particulièrement la méthode [Burguera et al., 2010]. Elle sera d'ailleurs testée dans le chapitre 4 en comparaison avec notre méthode. Comme la plupart des techniques décrites, elle fonctionne bien dans des conditions expérimentales favorables. Qu'en est-il dans un cadre péri-urbain ? De plus, elle ne prend pas en compte toutes les informations à sa disposition. Ainsi, lors de l'extraction des points qui vont servir à l'algorithme de *scan matching* (généralement ICP) les informations de niveaux sont supprimées. Ne serait-il pas important de considérer qu'un niveau de réflexion plus ou moins intense peut aussi fournir une information ? C'est ce que propose le chapitre 4 qui apporte une solution originale sur la mise en correspondance de *scan* radar par l'utilisation de la *Transformée de Fourier-Mellin*.

Pour finaliser les solutions choisies, rappelons que l'algorithme estimatif est l'EKF-SLAM. Ce choix a été fait, par l'équipe, au tout début des travaux. Cela interdit de choisir une représentation type grille d'occupation qui, faut-il le rappeler est la plus adaptée à la construction d'une cartographie riche en détails. Dans ce contexte, une technique s'inspirant des grilles d'occupation sera utilisée dans le seul but de créer une carte visuelle de l'environnement. Elle sera déterminée hors cadre SLAM et n'aura pour objectif que de fournir à la demande de l'opérateur, une représentation riche des lieux traversés.

Enfin, des tests de SLAM longues distances seront effectués. Dans le but de conserver un intérêt temps réel, ils seront réalisés avec des outils de programmation optimisés.

Le chapitre 4 va donc débiter par la présentation d'une nouvelle technique de *scan matching* : la mise en correspondance par *Transformée de Fourier-Mellin (FMT)*. Il présente ensuite une approche novatrice liée aux propriétés de la *FMT* pour déterminer la matrice de covariance associée à la transformation rigide estimée. Il se termine par la description de toutes les étapes de l'algorithme estimatif orienté EKF-SLAM.



# Chapitre 4

## Cartographie et localisation simultanées avec le radar K2Pi

### 1 Introduction

Le radar K2Pi est un radar tournant à modulation de type FMCW, c'est-à-dire à modulation de fréquence à onde continue. Il permet, dans des conditions atmosphériques sévères, de garantir l'observation de son environnement proche. Ses informations, fortement bruitées par des problèmes de saturation et de réflexions multiples, sont très denses. Elles doivent permettre d'obtenir une connaissance détaillée de l'environnement exploré.

Le chapitre précédent a montré que la littérature orientée SLAM offre plusieurs exemples d'utilisation d'un tel capteur. Néanmoins, ils correspondent à des conditions d'utilisation favorables, telles que le rajout d'amers artificiels ou le déplacement dans une zone sans objets mobiles, incompatibles avec les objectifs fixés dans ce document.

Le chapitre 2 a présenté le radar et a montré les méthodologies mises en œuvre pour créer, à partir de faisceaux, des panoramiques représentatifs de l'environnement. Ce mode de représentation « global » se retrouve dans la littérature SLAM lors de l'utilisation de *scans* laser ou de données sonars. Les auteurs proposent alors des techniques d'extraction des informations extéroceptives qui utilisent le *scan matching*. Chaque panoramique est considéré comme une carte locale qu'il suffit de comparer avec un panoramique précédent pour déterminer le déplacement effectué par le véhicule.

Ce chapitre a pour objectif de présenter tous les éléments mis en œuvre dans la création d'un SLAM utilisant le radar K2Pi comme capteur extéroceptif. La figure 4.1 donne une description des fonctionnalités utilisées. On retrouve un bloc d'entrée qui réalise la mise en correspondance des panoramiques et le calcul des incertitudes. Sa sortie fournit la transformation rigide entre les deux positions d'acquisition et les incertitudes sur la précision de cette mesure. Il y a ensuite l'algorithme estimatif EKF-SLAM qui calcule la nouvelle position et la matrice de covariance associée. Enfin, le dernier bloc calcule, connaissant les panoramiques et les positions associées, une reconstruction de l'environnement (cette carte est purement informative pour l'utilisateur).

La section 2 s'attache à présenter une toute nouvelle méthode de *scan matching*,

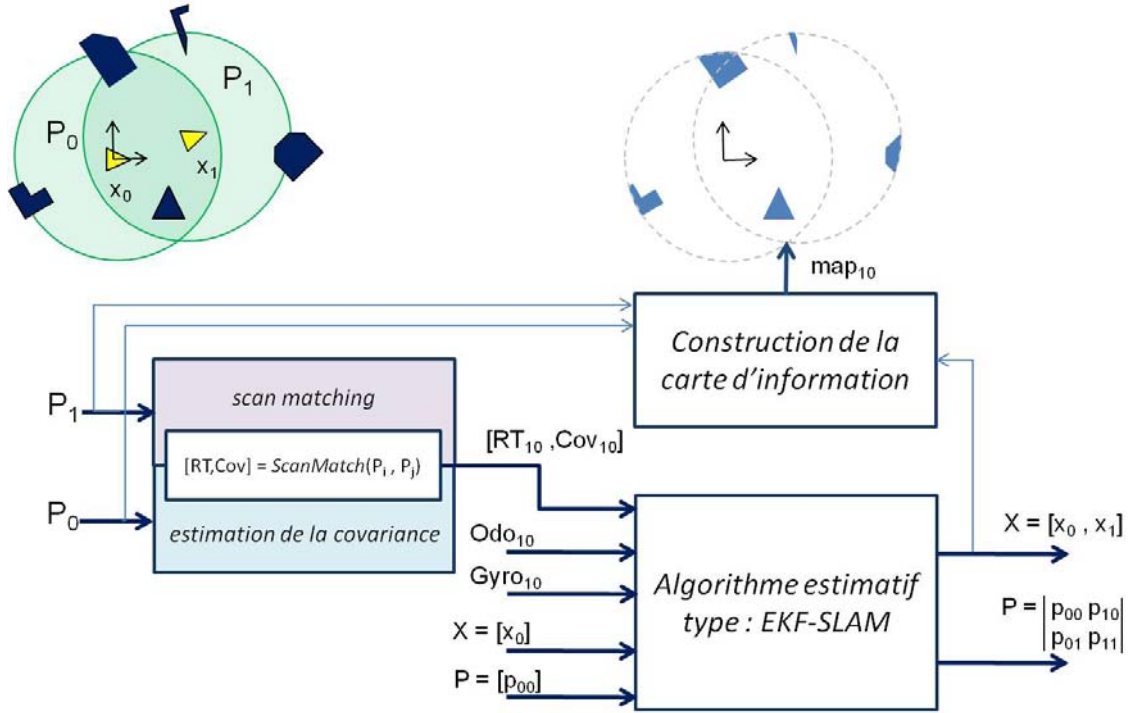


FIGURE 4.1 – Mise en œuvre d'un SLAM radar : connaissant, la position de départ  $x_0$ , les panoramiques  $P_0$  et  $P_1$ , les données proprioceptives du gyromètre ( $Gyro_{10}$ ) et de l'odomètre ( $Odo_{10}$ ), on peut déterminer la position  $x_1$ . On aura pour cela calculé la transformation rigide  $RT_{10}$  qui indique le déplacement effectué entre les deux positions, et les covariances associées (incertitude sur la mesure).

en marge de celles habituellement utilisées en SLAM, qui se fonde sur les propriétés de la *Transformée de Fourier-Mellin*. Cette technique fournit en résultat la transformation rigide  $RT$ .

Un des problèmes des algorithmes de *scan matching* est de devoir fournir, en complément de la transformée, une valeur d'incertitude sur le résultat. Cette valeur est ensuite directement intégrée dans la matrice de covariance de l'algorithme de SLAM. La section 3 présente les techniques les plus communément utilisées en SLAM et les complète par la description d'un nouveau procédé directement associé à la *Transformée de Fourier-Mellin*.

La section 4 donne une étude complète et la mise en œuvre d'un algorithme EKF-SLAM adapté à l'utilisation d'un radar K2Pi.

Enfin, le chapitre se conclut par un exemple d'application qui illustre l'intérêt de l'approche.

## 2 *Scan matching* par Transformée de Fourier-Mellin

### 2.1 Introduction

Si on observe un panoramique radar, on voit une carte d'intensités qui représente la zone entourant le radar. Les niveaux indiquent les réflexions plus ou moins importantes des rayons contre des obstacles. L'observation de deux panoramiques éloignés

de quelques mètres donne deux images différentes mais qui présentent néanmoins des zones communes identifiables.

Dans ce cadre, on peut voir dans la littérature orientée imagerie quelles techniques sont capables de mettre en correspondance deux images ainsi constituées. Ainsi, Brown et Basaeed [1992] réalisent une étude générale des différents besoins de mise en correspondance d'images et analysent les techniques les plus couramment utilisées.

On peut citer comme domaines d'application de ce procédé :

- l'imagerie pour la segmentation, la reconnaissance d'objets, la reconstruction de formes, le suivi, la stéréovision, la reconnaissance de caractères ;
- l'imagerie médicale [Guo et al., 2005], avec le diagnostic médical automatique (détection de tumeurs, localisation de maladies, la classification des cellules et des chromosomes) ;
- l'imagerie aérienne [Xie et al., 2003] pour des applications civiles, militaires, géologiques, océanographiques.

Dans tous ces domaines d'application, on retrouve différents paramètres à prendre en compte. Ils sont liés au capteur lui-même (bruit électronique, parasitage, limitations physiques, etc.) et à la scène observée.

En définitive, la littérature fournit trois méthodes principales pour la mise en correspondance d'images issues de caméras (de vision, infrarouges, tomographiques). Il y a les méthodes de corrélation (via Fourier) [Kuglin et Hines, 1975 ; Brigham, 1988 ; Reddy et Chatterji, 1996 ; Stone et al., 2001], les méthodes de mise en correspondance de points extraits, on retombe alors sur des méthodes type ICP [Lu et Milios, 1994 ; Fitzgibbon, 2001] et des méthodes de recherche de l'orientation par le calcul des directions des gradients [Fitch et al., 2002], enfin les méthodes de mise en correspondance de modèles déformables ou non.

Deux techniques prédominent : la corrélation fondée sur Fourier et les méthodes ICP. Le chapitre 3 a montré les limites de la méthode ICP et sa possible inadéquation pour la mise en correspondance de panoramiques radar.

Cette section va donc tout d'abord décrire un nouveau procédé de *scan matching*. Il utilise la *Transformée de Fourier-Mellin* et s'inspire des méthodes appliquées en imagerie traditionnelle. Puis, elle va réaliser un test comparatif sur des données réelles, avec la méthode ICP.

## 2.2 Principe

Le principe utilisé pour calculer une observation (une transformation rigide 2D  $[R, T]$ ) à partir de deux panoramiques radar, doit respecter certaines hypothèses :

- l'environnement est plan et non déformable ;
- l'environnement ne doit pas être (ou peu) modifié entre deux enregistrements ;
- le nombre d'objets mobiles doit être faible et négligeable par rapport au reste des éléments observés ;
- il ne doit pas y avoir de changement d'échelle entre les deux observations ;
- on doit respecter des temps de calculs faibles en vue d'une implémentation temps réel.

Afin d'effectuer la mise en correspondance, on utilise la méthode de la *Transformée de Fourier-Mellin* [Brown et Basaeed, 1992 ; Kuglin et Hines, 1975] qui fait



partie des techniques utilisant la *cross-correlation*.

Ces techniques diffèrent des méthodes issues d'ICP par le fait qu'elles minimisent les distances entre des points mis en correspondance. Seule la position des points (lignes, surfaces) compte. C'est d'ailleurs pour cela que l'association des points entre images s'avère parfois délicate. L'ajout d'informations issues d'autres éléments du processus (autre capteur, informations estimatives, etc.) est souvent réalisé pour résoudre ces associations.

Dans le cadre de la *cross-corrélation*, on considère que l'intensité de réflexion d'un objet change peu entre deux positions d'acquisitions. Ainsi, si deux observations (par exemple issues d'une caméra) d'une même scène sont réalisées, avec un léger déplacement, les variations d'intensité seront semblables entre les deux jeux de données. On recherche donc la mise en correspondance des images en comparant les niveaux de gris.

Cela revient à minimiser la distance euclidienne entre deux observations  $f_1(x, y)$  et  $f_2(x - u, y - v)$  d'une même scène après un déplacement  $(u, v)$ .

$$\begin{aligned} d_{f_1 f_2}^2(u, v) &= \sum_{x, y} [f_1(x, y) - f_2(x - u, y - v)]^2 \\ &= \sum_{x, y} [f_1^2(x, y) + f_2^2(x - u, y - v) - 2f_1(x, y)f_2(x - u, y - v)] \end{aligned} \quad (4.1)$$

Les deux premiers termes sont presque constants si l'éclairage de l'objet est uniforme. Ils ne sont liés qu'à  $f_1$  ou à  $f_2$ , et non aux différences qui apparaissent lorsque  $f_1$  se transforme en  $f_2$ . Pour cette raison, on estime la similitude entre deux images par le terme qui reste. Il indique la corrélation entre les deux images :

$$C(u, v) = \sum_{x, y} f_1(x, y)f_2(x - u, y - v) \quad (4.2)$$

Dans ce contexte, la transformée de Fourier dispose de plusieurs propriétés qui peuvent être employées pour ce type de recherche : translation, rotation, réflexion, échelle. De plus, une implantation du type *Fast Fourier Transform* (FFT) permet d'obtenir des temps de calculs très courts.

### 2.2.1 Calcul d'une translation

Le calcul d'une translation entre deux images décalées peut être effectué par une *corrélation de phase* [Kuglin et Hines, 1975]. Son principe est le suivant :

étant données deux images  $I_1$  et  $I_2$  qui diffèrent simplement par une translation  $(\Delta x, \Delta y)$ , i.e.

$$I_2(x, y) = I_1(x - \Delta x, y - \Delta y) \quad (4.3)$$

On a alors dans le domaine fréquentiel :

$$F_2(w_x, w_y) = F_1(w_x, w_y).e^{-j(w_x \Delta x + w_y \Delta y)} \quad (4.4)$$

Les deux images ont le même module mais une phase différente directement proportionnelle au déplacement. Avec une représentation exponentielle  $F(\vec{w}) = |F|.e^{j\phi(\vec{w})}$ , la différence de phase est alors donnée par  $e^{j(\phi_1 - \phi_2)}$ .

On définit la corrélation de phase dans le domaine fréquentiel par :

$$Corr(w_x, w_y) = e^{(w_x \Delta x + w_y \Delta y)} = \frac{F_1(w_x, w_y) F_2^*(w_x, w_y)}{|F_1(w_x, w_y) F_2^*(w_x, w_y)|} \quad (4.5)$$

avec  $*$  le conjugué complexe.

La transformée de Fourier inverse de l'équation (4.5) fournit un pic centré sur la valeur de translation entre les deux images.

On définit alors la corrélation dans le domaine temporel par :

$$Corr(x, y) = TF^{-1}(Corr(w_x, w_y)) = \delta(x + \Delta x, y + \Delta y) \quad (4.6)$$

et donc :

$$(\Delta x, \Delta y) = \arg \max_{(x, y)} \{Corr(x, y)\} \quad (4.7)$$

### 2.2.2 Rotation et homothétie

L'utilisation de la *Transformée de Fourier-Mellin* permet de calculer les paramètres d'une similitude (rotation et homothétie) comme un vecteur de translation, grâce à une représentation log-polaire de l'espace des fréquences  $(u, v) \rightarrow (\theta, \log \rho)$ .

Si on prend l'exemple précédent, cette fois l'image  $I_2$  est la transformée de  $I_1$  par une rotation d'angle  $\alpha$ , une homothétie de rapport  $\sigma$ , et une translation de vecteur  $(\Delta x, \Delta y)$ . On a alors :

$$I_2(x, y) = I_1(\sigma(x \cos \alpha + y \sin \alpha) - \Delta x, \sigma(-x \sin \alpha + y \cos \alpha) - \Delta y) \quad (4.8)$$

Les amplitudes des transformées de Fourier de  $I_1$  et  $I_2$  sont liées par la relation suivante :

$$|F_2(u, v)| = \frac{1}{\sigma^2} \left| F_1\left(\frac{1}{\sigma}(u \cos \alpha + v \sin \alpha), \frac{1}{\sigma}(-u \sin \alpha + v \cos \alpha)\right) \right| \quad (4.9)$$

L'équation (4.9) montre que l'amplitude :

- ne dépend pas de la translation  $(\Delta x, \Delta y)$  ;
- subit une rotation d'angle  $\alpha$  ;
- subit une modification d'échelle d'un facteur  $\frac{1}{\sigma}$ .

En passant les fréquences en coordonnées polaires :

$$\begin{aligned} F_{p1}(\theta, \rho) &= |F_1(\rho \cos \theta, \rho \sin \theta)| \\ F_{p2}(\theta, \rho) &= |F_2(\rho \cos \theta, \rho \sin \theta)| \end{aligned}$$

et en passant des coordonnées radiales en coordonnées logarithmes :

$$\begin{aligned} r &= \log \rho & F_{lp1}(\theta, r) &= F_{p1}(\theta, \rho) \\ s &= \log \sigma & F_{lp2}(\theta, r) &= F_{p2}(\theta, \rho) \end{aligned}$$

on obtient finalement :

$$F_{lp2}(\theta, r) = \frac{1}{\sigma^2} \cdot F_{lp1}(\theta - \alpha, r - s) \quad (4.10)$$

On retrouve ainsi une similitude avec une translation dans l'espace temporel (équation (4.3)). On peut donc utiliser les équations (4.5), (4.6), (4.7) pour déterminer les valeurs de l'angle  $\alpha$  et du facteur d'échelle  $s$ .

Dans notre application, le radar ne provoque pas de changement d'échelle, on peut alors supprimer l'étape de transformation en log-polaire et réaliser simplement une transformation polaire. La figure 4.2 montre l'implantation de cette technique pour la recherche de la transformation rigide entre deux panoramiques radar.

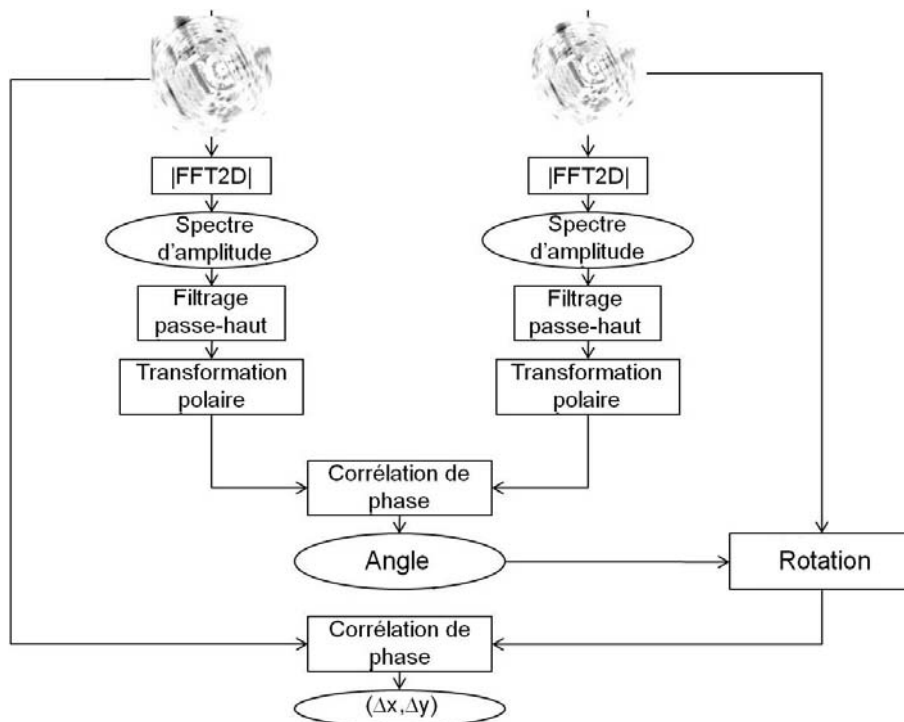


FIGURE 4.2 – Algorithme de Fourier-Mellin pour la recherche de la transformation rigide 2D  $[R, T]$ .

Les bases mathématiques posées, la méthode par *Transformée de Fourier-Mellin* (FMT) peut maintenant être validée sur des images de tests puis sur des panoramiques radar.

## 2.3 Mise en œuvre et validation

### 2.3.1 Un espace discret

L'algorithme *FMT* va être appliqué sur des images, donc sur un espace discret. La littérature conseille, afin de limiter les effets de recouvrement de spectre, d'appliquer un fenêtrage. Par exemple, on peut appliquer une fenêtre de *Hanning* sur l'image avant d'effectuer la transformée [Gonzalez et Woods, 2001].

De plus (cf. figure 4.2), lors du passage en coordonnées polaires, des problèmes d'interpolation apparaissent (cf. figure 4.3). Divers auteurs se sont penchés sur ce problème [Lawton, 1988 ; Wolberg et Zokai, 2000 ; Keller et al., 2005]. Il en ressort que toutes ces méthodes améliorent la précision des résultats au détriment d'une augmentation des temps de calculs. Cela est intéressant lorsque l'on souhaite passer à une précision sub-pixelique. Il faut dans ce cas, des données très précises ce qui n'est

pas le cas des panoramiques radar. En conséquence, une interpolation du premier ordre, rapide et suffisamment précise, est utilisée dans notre contexte.

On peut, afin d'augmenter la résolution, facilement compléter les images avec des zéros afin d'augmenter la taille de la *FFT* [Castro et Morandi, 1987].

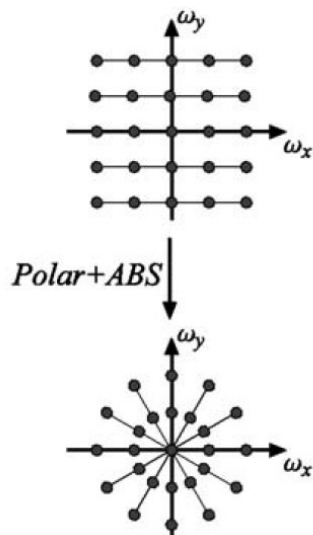


FIGURE 4.3 – Transformation de  $|F(w_x, w_y)|$  en  $F(\theta, \rho)$ .

### 2.3.2 Essai sur une image artificiellement translatée

Les premiers essais ont été réalisés sur des images artificiellement translatées et ré-orientées. La figure 4.4 montre un test réalisé sur « *Lena* », image maintes fois utilisée en imagerie. Les conditions de tests sont :

- une image de référence de  $256 \times 256$  pixels et de 256 niveaux de gris ;
- une imagerie, qui est une partie de l'image originale décalée et orientée.

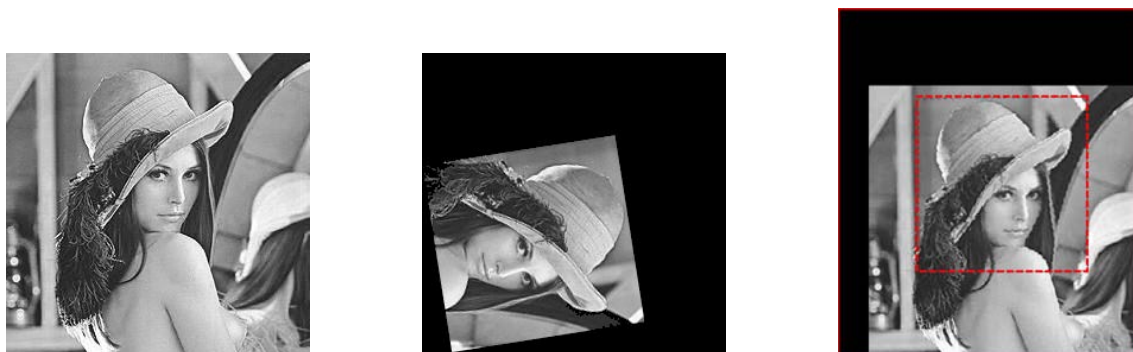


FIGURE 4.4 – Superposition automatique de deux images.

On peut voir sur la figure 4.5, le pic de corrélation qui permet de déterminer les paramètres de translation  $(\Delta x, \Delta y)$ . Ils sont calculés après avoir orienté la deuxième image dans le repère de la première (angle de ré-orientation déjà déterminé, voir figure 4.2).

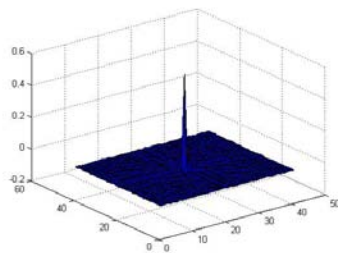


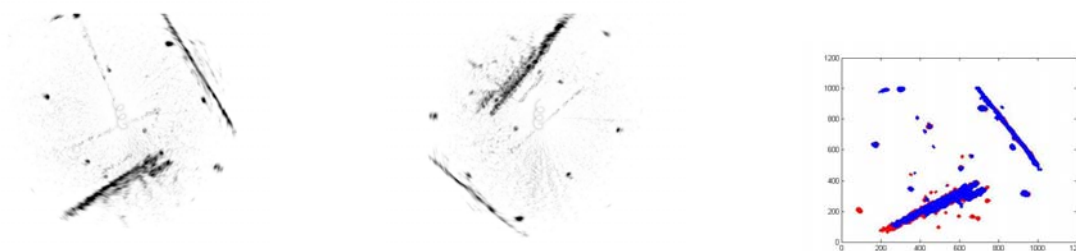
FIGURE 4.5 – Pic de corrélation représentant les valeurs de translation.

### 2.3.3 Essai sur des panoramiques

L'objectif est de faire fonctionner cet algorithme sur des données radar. La figure 4.6 montre un exemple de détermination de la transformation rigide 2D  $[R, T]$  entre deux acquisitions. Elles ne sont pas tout à fait identiques et chacune dispose d'informations absentes chez son homologue.

Les conditions d'expérimentations sont :

- images de  $1024 \times 1024$  pixels et 256 niveaux de gris ;
- images seuillées pour éliminer les informations de bas niveau ;
- *FFT* effectuées sur 1024 pixels. La résolution en retour de *FFT* :
  - pour la translation : 20 cm / pixel ;
  - pour la rotation :  $0.35^\circ$  / pixel.


 FIGURE 4.6 – Superposition automatique de deux images après détermination de la transformation rigide  $[R, T]$ .

On peut voir sur la figure 4.7, le pic de corrélation qui permet de déterminer la translation  $(\Delta x, \Delta y)$ . Par rapport au pic de corrélation trouvé sur l'exemple précédent, on s'aperçoit que le pic est fondu dans un bruit important (voir section 2.4).

## 2.4 Bruit sur la corrélation de phase

Dans le cas de panoramiques, on s'aperçoit que la corrélation de phase est fortement bruitée. Cela n'empêche pas de déterminer la valeur de translation mais on voit clairement apparaître les limites de son utilisation.

Plus les images vont différer, et plus la corrélation sera difficile à réaliser. Ce constat logique peut être complété par l'observation suivante. La formule de détermination du pic dans le domaine fréquentiel (équation (4.5)) est normalisée. Dans

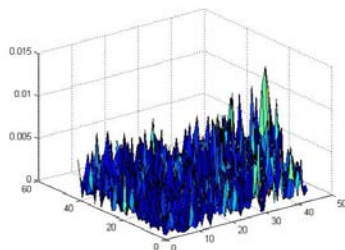


FIGURE 4.7 – Pic de corrélation issu de la comparaison de deux panoramiques.

le cas de la comparaison de deux images très ressemblantes, cela ne pose pas de problème, bien au contraire, cela résout les problèmes de changement d'éclairage. L'équation (4.6) fournit alors les valeurs de translation. Même si l'on bruite les images (par exemple avec un bruit blanc), on arrive à retrouver les valeurs de translation (voir figure 4.8).

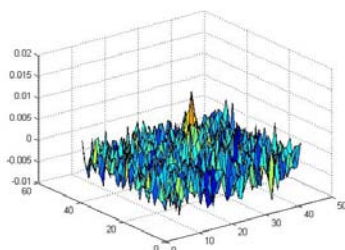


FIGURE 4.8 – Pic de corrélation après bruitage des images Lena avec un bruit blanc de variance 1.

Ce qui pose réellement problème, c'est lorsque les deux images possèdent peu d'informations avec quelques zones communes et beaucoup de zones dissemblables. On peut voir un exemple dans la figure 4.9 où la valeur trouvée ne correspond pas à la bonne superposition.

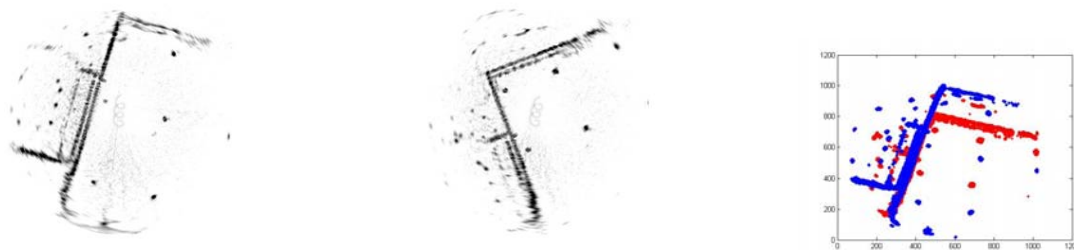


FIGURE 4.9 – La mise en correspondance est fausse. Les images contiennent de nombreuses différences qui mettent en échec l'algorithme.

Si l'on observe la corrélation de phase (voir figure 4.10), on voit que le bruit est très important et que le choix du pic de plus grande amplitude devient délicat à effectuer. Cela est causé, d'une part, par le fait que l'on ne dispose que d'un nombre restreint d'informations communes et, d'autre part, par le fait que l'équation (4.5)

qui normalise le résultat, provoque une amplification très importante du bruit. La division par  $|F_1(w_x, w_y)F_2^*(w_x, w_y)|$  peut générer de fortes amplifications par la prise en compte de fréquences parasites.

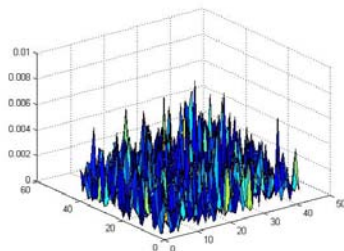


FIGURE 4.10 – Pic de corrélation entre deux panoramiques qui ne possèdent que peu d'informations communes.

Si on augmente graduellement le nombre d'éléments communs (pour cela on élimine petit à petit de la première image les zones qui divergent de la seconde) le signal informatif devient plus important et la superposition devient possible. La figure 4.11 le montre : le bruit reste mais la zone commune augmente. On peut dire que cela correspond aux limites de la méthode.

En imagerie, une forte ressemblance existe entre les vues, et les zones communes y sont nombreuses. Ainsi cette technique fonctionne parfaitement. En revanche, avec des panoramiques radar, le bruit est important et la ressemblance entre les vues peut être faible. De plus, les panoramiques peuvent ne disposer que de peu d'informations. On est ainsi rapidement limité. Il faut donc améliorer le procédé. C'est ce qui est expliqué dans la section suivante.

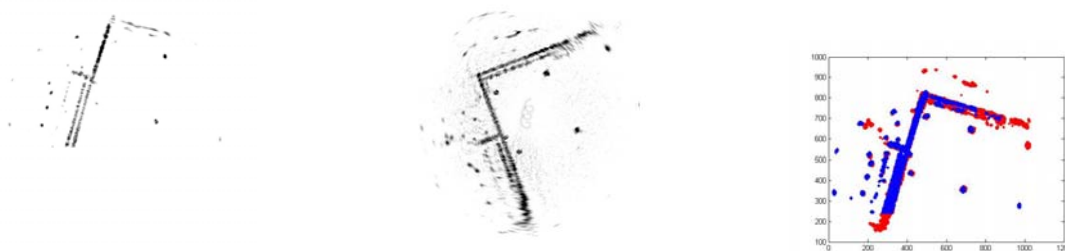


FIGURE 4.11 – On supprime artificiellement de l'image de gauche la zone non commune. L'effet est immédiat, le calcul de la superposition fonctionne.

## 2.5 Corrélation

Afin d'améliorer l'algorithme de recherche des transformations rigides entre panoramiques bruités et dissemblables, nous allons supprimer de nos équations la normalisation. Le but est de corriger l'effet d'amplification du bruit. L'équation (4.5) va devenir :

$$Corr(w_x, w_y) = F_1(w_x, w_y)F_2^*(w_x, w_y) \quad (4.11)$$

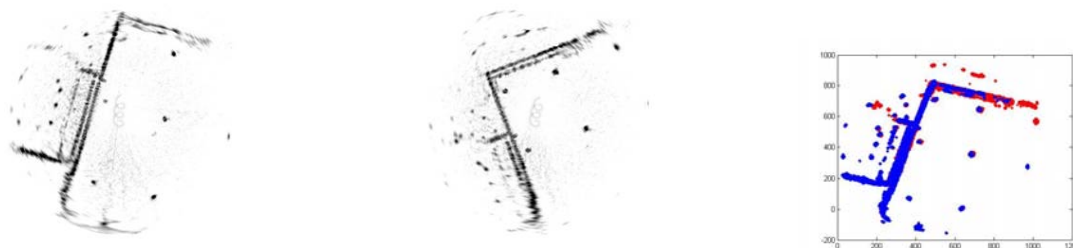


FIGURE 4.12 – La superposition est correcte grâce à l'utilisation de l'équation (4.11) pour déterminer la transformation rigide  $[R, T]$ .

Cela donne pour l'exemple précédent une superposition correcte (cf. figure 4.12).

Le résultat de corrélation dans le domaine temporel est représenté dans la figure 4.13.

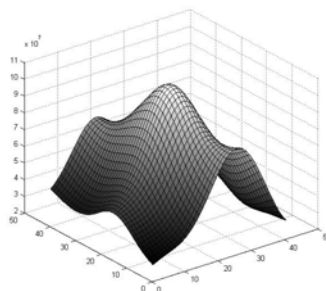


FIGURE 4.13 – Corrélation par la formule 4.11. L'extraction du maximum devient aisée.

On passe d'une *corrélation de phase* à une corrélation traditionnelle. Une des propriétés de la méthode qui était son insensibilité aux fortes variations d'éclairage, se réduit. Néanmoins, dans le cadre de panoramiques radar, cela ne pose pas un réel problème puisque les variations sont faibles.

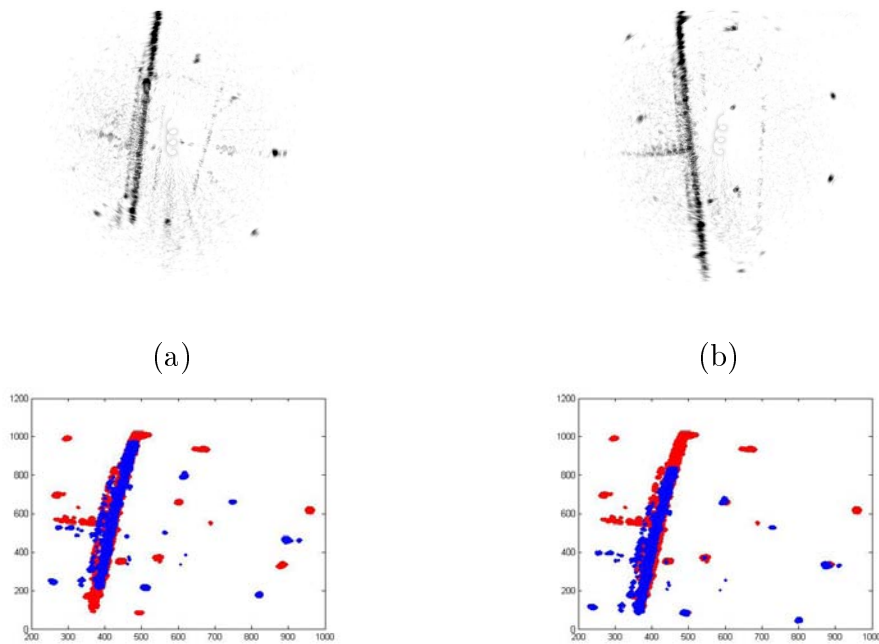
### 2.5.1 Limites

Le coefficient de corrélation trouvé à l'issue de la mise en correspondance peut être utilisé comme coefficient de robustesse, c'est-à-dire, comme un indicateur qui peut renseigner l'utilisateur sur la qualité de la mise en correspondance. Il faut néanmoins pondérer cette affirmation.

En effet, une transformation rigide  $[R, T]$  peut être trouvée si les deux images possèdent des zones communes et si l'environnement est suffisamment riche en information afin que l'on puisse en extraire, sans ambiguïté, la valeur de déplacement.

Si l'on regarde l'exemple de la figure 4.14, on a deux panoramiques qui sont acquis à très peu de distance l'un de l'autre (10 m). La zone est pauvre en information, la mise en correspondance trouvée est alors fautive. Le coefficient de robustesse fournit une valeur élevée qui ne reflète pas l'échec de la mise en correspondance. Il n'est donc pas fiable dans le cas présent.





(la superposition est mauvaise)    (la superposition qui aurait dû être trouvée)

FIGURE 4.14 – La faible quantité d’information contenue dans les panoramiques (a) et (b) ne permet pas de réaliser un appariement correct.

Si on prend l’exemple de la figure 4.15, la valeur de la transformation 2D  $[R, T]$  est fautive. Le résultat de la corrélation fournit un coefficient de robustesse faible qui indique le mauvais appariement (voir [Brown et Basaeed, 1992]). Dans ce cas, le coefficient peut écarter ce résultat de la liste des résultats valides.

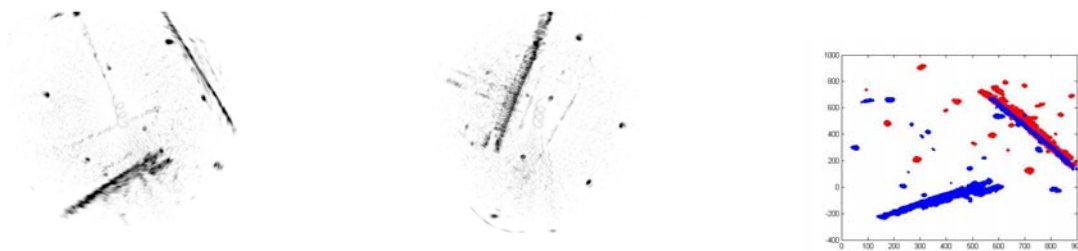


FIGURE 4.15 – La faible quantité d’information dans les panoramiques crée une ambiguïté forte qui donne un mauvais résultat d’appariement. Ici, le coefficient de robustesse l’indique par une valeur faible.

### 2.5.2 Conclusion

Si le coefficient de robustesse fournit une valeur faible, il est certain que le calcul de la transformation rigide 2D  $[R, T]$  n’est pas fiable. Il ne devra pas être utilisé dans l’algorithme de SLAM.

Cependant, si ce coefficient est bon, cela ne signifie pas obligatoirement que l’appariement est correct d’où la mise en œuvre de critères complémentaires dans

l'algorithme de SLAM.

## 2.6 Filtre anti-repliement

Comme expliqué dans la section 2.3.1, le passage à un espace fini crée au niveau du calcul de la FFT des déformations liées au repliement du spectre. Faut-il dans ce cas, prévoir la mise en place d'un correctif ?

Pour réduire ce phénomène, la littérature conseille de mettre en œuvre un fenêtrage temporel qui réduit l'amplitude des informations situées aux extrémités du signal afin d'amoindrir les effets de bord.

Dans notre application, cela a pour effet de réduire l'amplitude des informations situées en périphérie de l'image. Les essais réalisés ont permis de faire différentes constatations, qui sont :

- une limitation des déformations ;
- pas d'amélioration en ce qui concerne la recherche des transformations rigides (sans le fenêtrage il y a bien une déformation du spectre, mais elle n'empêche pas, à terme, de retrouver la valeur maximum de corrélation) ;
- une réduction importante des possibilités de détection des appariements entre panoramiques éloignés. Cela est lié à la diminution des informations de périphérie.

En conclusion, ce filtre permet de réduire les déformations liées au repliement de spectre mais dans le même temps, il réduit l'efficacité du procédé de mise en correspondance. Il ne sera pas mis en œuvre dans l'application.

## 2.7 Comparaison avec une mise en correspondance de type ICP

### 2.7.1 Introduction

La méthode ICP (et ses variantes) représente la technique la plus en vogue dans le *scan matching*. Elle offre la possibilité, avec des temps de calcul faibles, de déterminer la transformation rigide entre deux flux de données acquis à des instants différents. Elle nécessite de disposer d'une valeur d'initialisation proche de la valeur finale (afin d'éviter de tomber dans des minima locaux). Son application la plus traditionnelle se fait sur des capteurs extéroceptifs de type imageur laser.

Dans l'exemple de la figure 4.16, le capteur fournit un flot de points qui représente la distance aux obstacles pour une orientation donnée. Les impacts sont des informations de distances par rapport au véhicule. Elles sont précises et souffrent peu des phénomènes de speckle (cf. figure 4.17) et de réflexion des panoramiques radar.

L'utilisation en 2D de l'algorithme ICP nécessite de disposer d'informations qui représentent la localisation des obstacles. Son application sur des panoramiques radar va donc demander un traitement préliminaire afin d'extraire des amers caractéristiques utilisables.

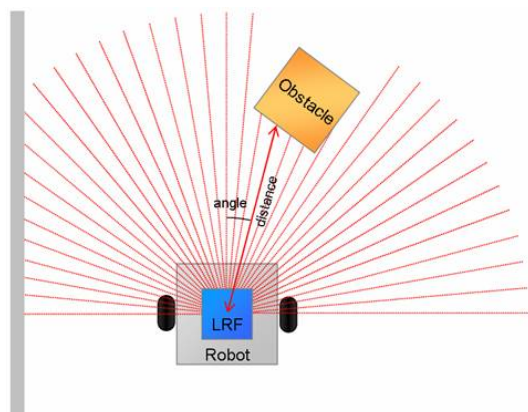


FIGURE 4.16 – Robot équipé d'un scan laser.

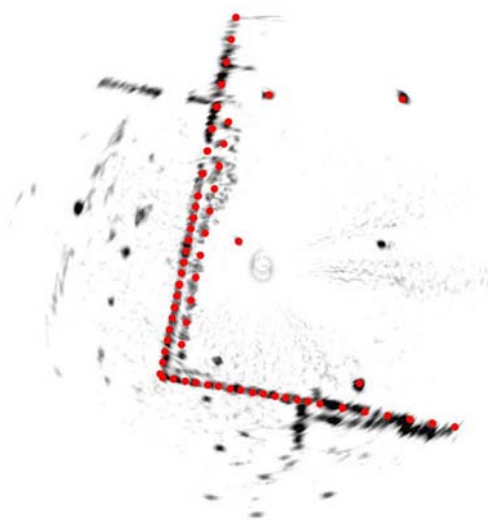


FIGURE 4.17 – Superposition d'une détection par laser (en rouge) et d'un panoramique radar (en niveaux de gris inversés).

### 2.7.2 Mise en forme des panoramiques

La mise en œuvre de l'algorithme ICP est effectuée sur des données préalablement modifiées. Burguera et al. [2010] proposent d'extraire les données issues d'un sonar par un seuillage adaptatif (fonction de l'histogramme), suivi d'une érosion pour éliminer les informations non caractéristiques. Une fois ceci réalisé l'algorithme ICP peut être appliqué.

**Seuillage adaptatif** : l'extraction d'information dans un panoramique nécessite de le seuiller correctement. Un seuil fixe risquerait en effet d'aboutir à la perte d'information. Dans ce contexte, Burguera et al. [2010] proposent un seuillage adaptatif fonction de l'histogramme sur les données. Une fois ce dernier calculé, on détermine la densité de probabilité suivant une loi normale centrée réduite. On fixe le seuil à 99% de la densité de probabilité, soit dans l'exemple de la figure 4.18, un seuillage

à 90.

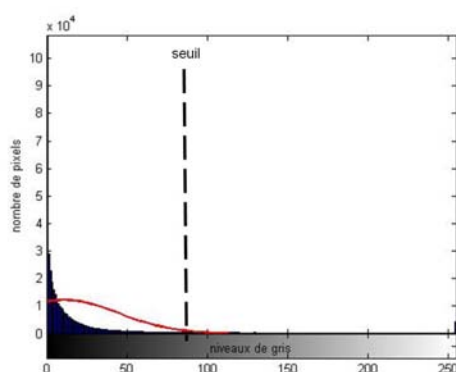


FIGURE 4.18 – Histogramme du panoramique 4.17.

La figure 4.19 représente le panoramique après seuillage. Il reste à réaliser un filtrage morphologique de type érosion pour éliminer les éléments peu significatifs.

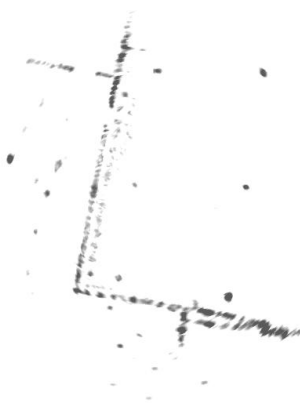


FIGURE 4.19 – Panoramique radar après seuillage.

**Érosion** : l'élimination du bruit résiduel se fait par un filtre morphologique de type érosion (via un disque de 4 pixels). L'image est ensuite binarisée. La figure 4.20 montre le panoramique après le traitement.

Une fois les traitements terminés, les données radar peuvent être utilisées pour le *scan matching* via ICP.

### 2.7.3 Comparatif *scan matching* ICP et FMT

La méthodologie utilisée pour comparer les deux techniques est décrite dans l'algorithme 5.

**Comparaison sans ajout d'erreurs** : la figure 4.21 montre les erreurs trouvées lors de 30 *scan matching* réalisés avec la méthode FMT. La figure 4.22 réalise la même opération avec la méthode ICP. Sur une transformation rigide  $[R, T]$ , les



FIGURE 4.20 – Panoramique radar après le filtrage morphologique de type érosion et la binarisation.

erreurs se situent au niveau de la translation (erreurs en  $X$  et  $Y$ ) et de la rotation (erreur en angle :  $\theta$ ).

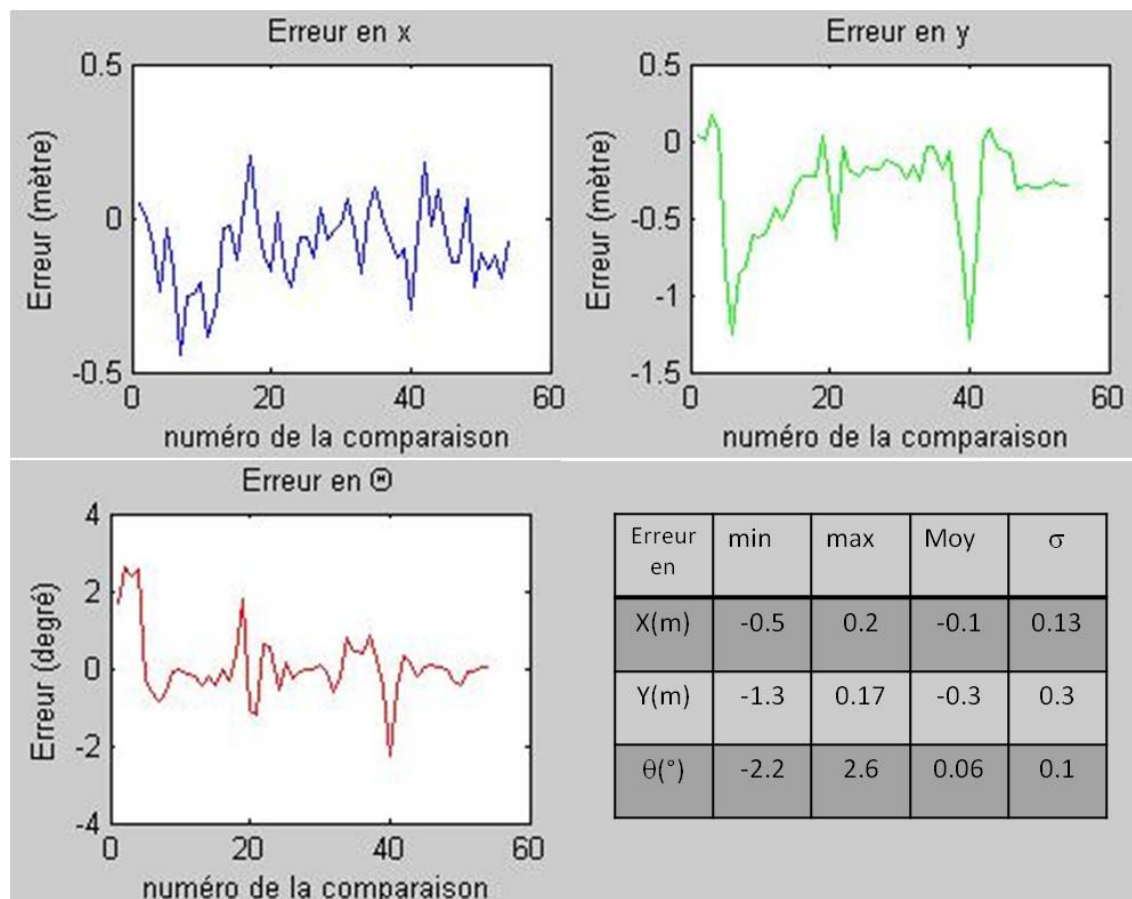


FIGURE 4.21 – Transformations rigides trouvées via FMT.

Les erreurs sont plus importantes par la méthode ICP même si elles sont relativement proches. Au niveau des temps de calculs, en moyenne, il faut compter (sous Matlab, avec une CPU 2 GHz) 3 s avec FMT et 2.8 s avec ICP.

**Algorithme 5:** Algorithme de comparaison des *scan matching* ICP et FMT

Pour N positions radar, acquisition des panoramiques et des données proprioceptives associées ;

**pour**  $i$  variant de 2 à N **faire**

Prédiction de  $\mathbf{x}_{v_i}$ , connaissant  $\mathbf{x}_{v_{i-1}}$  et les proprios  $\mathbf{Proprio}_i$  :

$\mathbf{x}_{v_i} = f(\mathbf{x}_{v_{i-1}}, \mathbf{Proprio}_i)$  ;

**pour**  $j$  variant de 1 à  $i-1$  **faire**

**si**  $Distance(\mathbf{x}_{v_i}, \mathbf{x}_{v_j}) < Distance_{max}$  **alors**

Estimation de la transformation rigide :

$[\mathbf{R}_{i,j}, \mathbf{T}_{i,j}] = g(\mathbf{x}_{v_i}, \mathbf{x}_{v_j})$  ;

Calcul des *scan matching* :

$[\mathbf{R}_{ICP}, \mathbf{T}_{ICP}] = ScanMatchICP(\mathbf{Pano}_i, \mathbf{Pano}_j, \mathbf{R}_{i,j}, \mathbf{T}_{i,j})$  ;

$[\mathbf{R}_{FMT}, \mathbf{T}_{FMT}] = ScanMatchFMT(\mathbf{Pano}_i, \mathbf{Pano}_j)$  ;

Calcul de la transformation rigide référence via l'utilisation des données GPS :

$[\mathbf{R}_{GPS_{i,j}}, \mathbf{T}_{GPS_{i,j}}] = g(\mathbf{x}_{GPS_i}, \mathbf{x}_{GPS_j})$  ;

Calcul des erreurs ;

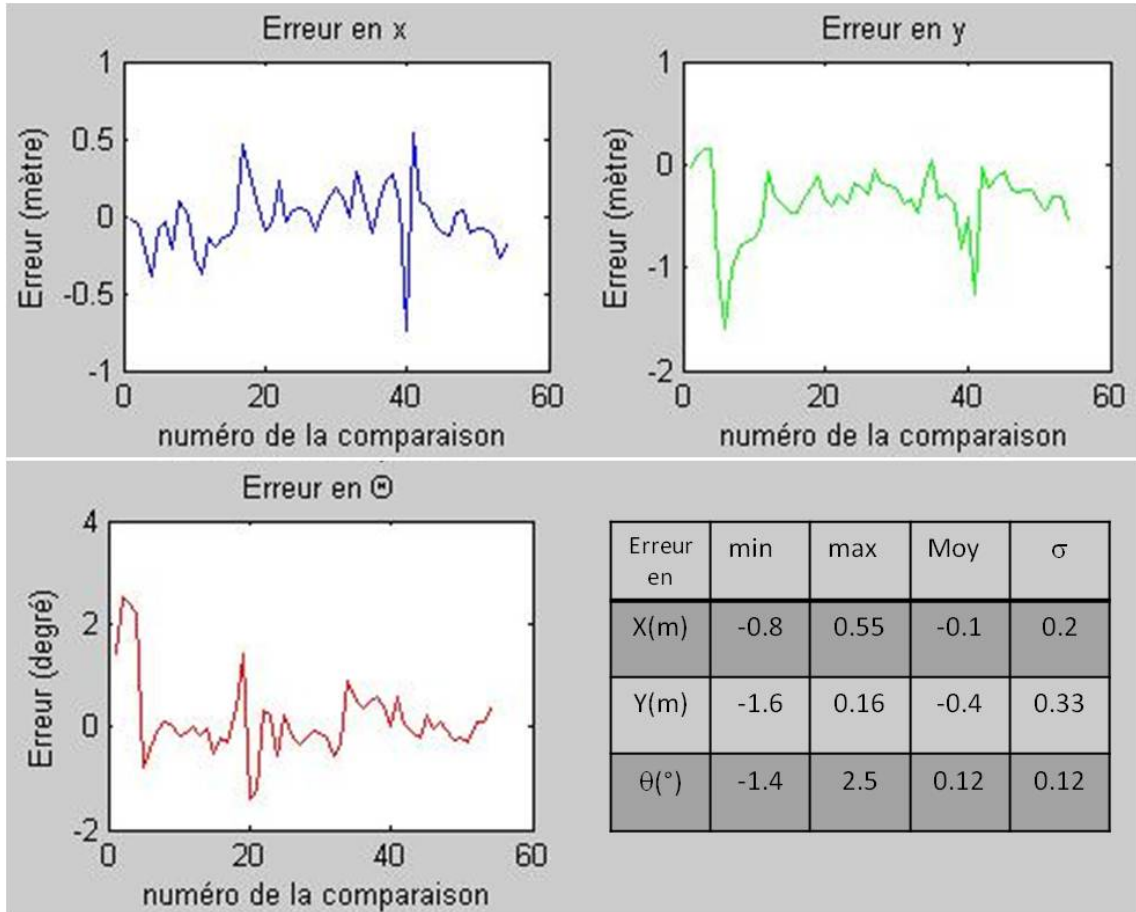


FIGURE 4.22 – Transformations rigides trouvées via ICP.

En conclusion, les méthodes fournissent des résultats relativement analogues. Il

faut néanmoins signaler, d'une part que ces tests ont été effectués sur des campagnes d'acquisition favorables (contexte péri-urbain qui offre beaucoup d'informations tel que le montre la figure 4.17) et, d'autre part, que ICP doit disposer de données d'initialisation correctes pour aboutir à ce résultat.

Le paragraphe suivant teste l'ICP dans des conditions d'utilisation plus sévères. Pour cela on va bruite artificiellement la valeur d'initialisation afin de vérifier sa robustesse.

**Comparaison avec ajout d'erreurs :** un des points sensibles de l'algorithme ICP se situe dans l'utilisation d'une valeur initiale. Cette dernière doit être la plus proche possible du résultat final afin d'éviter de tomber dans des minima locaux. La figure 4.23 montre les résultats trouvés pour la méthode ICP avec un ajout de bruit sur les capteurs proprioceptifs. Le bruit blanc gaussien a un écart-type de  $\sigma_{Odo} = 5\text{ m}$  pour l'odomètre et de  $\sigma_{Gyro} = 5^\circ$  pour le gyromètre. Le *scan matching* FMT ne dépend pas de ces paramètres d'initialisation, on retrouve donc les résultats présentés dans la figure 4.21.

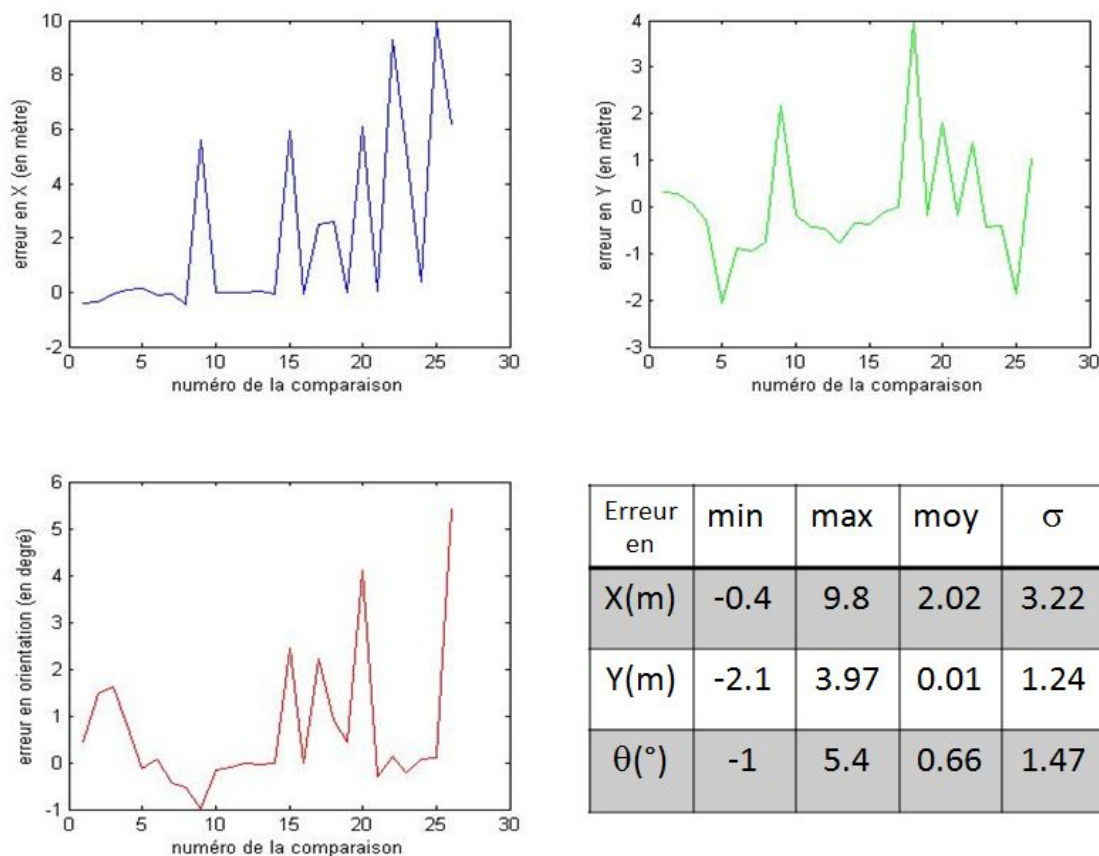


FIGURE 4.23 – Transformations rigides trouvées via ICP. La valeur d'initialisation est bruitée (les proprios utilisés pour calculer les valeurs d'initialisation ont subi un ajout de bruit d'écart type  $\sigma_{Odo} = 5\text{ m}$  pour l'odomètre et de  $\sigma_{Gyro} = 5^\circ$  pour le gyromètre).

On note que le *scan matching* par ICP est mis en échec lors du calcul des mises en correspondance : 9, 15, 20, 22, 25. La valeur d'initialisation est alors trop éloignée

du résultat final, l'algorithme tombe alors dans un minimum local. La figure 4.24-a) montre un exemple où l'initialisation ne permet pas d'aboutir à un calcul correct de la transformation rigide. Le scan destination, après transformation, ne se superpose pas au scan de référence. La figure 4.24-b) montre un cas où malgré une initialisation lointaine, l'ICP détermine bien la bonne transformation rigide. Il n'est pas tombé dans un minimum local.

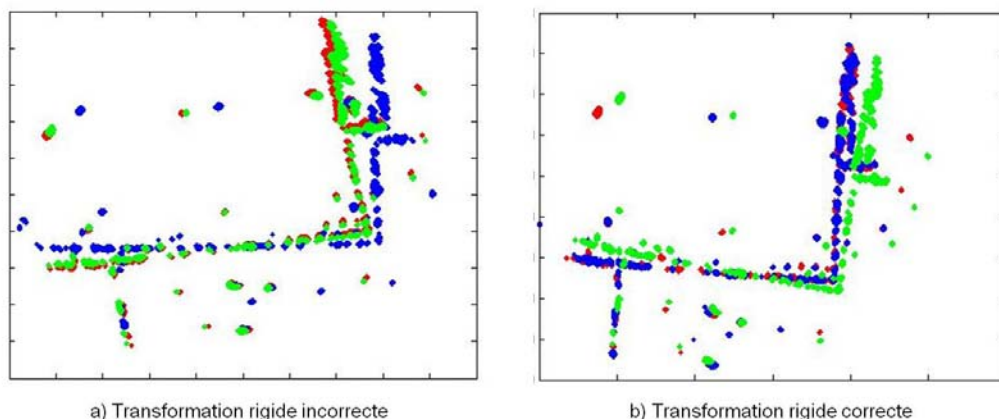


FIGURE 4.24 – Superposition de deux scans après transformation rigide. En bleu le scan de référence, en vert la position d'initialisation, en rouge le scan qui doit venir se superposer à la référence (si la transformation rigide est correcte).

En conclusion, une initialisation imprécise peut provoquer une divergence d'ICP. La méthode FMT procure de meilleurs résultats et est plus robuste dans les cas plus perturbés qui mettent en échec l'ICP. Elle ne nécessite pas une estimation *a priori* de la transformation rigide. Ses temps de calculs sont semblables à ceux d'ICP. Elle apparaît, dans le cadre d'un *scan matching* sur panoramiques radar, être une méthode parfaitement adaptée.

## 2.8 Conclusion

Cette section vient de présenter une nouvelle méthode de mise en correspondance de données extéroceptives issues d'un radar FMCW.

Sa comparaison avec la méthode classiquement utilisée, appelée ICP, montre sa robustesse et sa facilité d'utilisation. Elle ne demande aucune valeur d'initialisation et aucune opération préalable d'extraction des informations au sein des panoramiques.

Une fois la transformation rigide entre deux images déterminée, le résultat peut être utilisé dans l'algorithme d'estimation des positions successives. Ce dernier, afin d'améliorer son estimation, utilise une matrice de covariance associée à chaque observation. Cela représente une information sur l'incertitude de mesure. De cette façon, il peut fournir une estimation des positions en indiquant un degré de précision.

On peut, pour chaque observation (dans notre cas la transformation rigide trouvée) fixer une incertitude constante. Déterminée après des essais expérimentaux, elle représente une précision moyenne du résultat. Il apparaît néanmoins intéressant de pouvoir calculer l'incertitude pour chaque cas, de cette façon l'estimation



des positions n'en sera que meilleure. C'est ce que propose de détailler la section suivante.

## 3 Estimation des incertitudes sur la transformation rigide

### 3.1 Introduction

Une fois la transformation rigide déterminée, il est important de pouvoir estimer les incertitudes liées à la mise en correspondance de deux images.

La littérature scientifique s'attache à la recherche des erreurs liées aux techniques de calculs et aux approximations numériques qui en découlent. La recherche des incertitudes liées au caractère bruité des données entrantes est rarement traitée. C'est pourtant un point important dans le cadre du *scan matching* où chaque information issue d'un capteur doit être associée à sa matrice de covariance afin de pouvoir faire évoluer au mieux le filtre probabiliste d'estimation des positions.

Dans le cadre de la mise en correspondance par la *Transformée de Fourier-Mellin*, les erreurs sur la transformation rigide sont causées par des erreurs numériques et par les erreurs liées au capteur. La littérature s'attarde largement sur les problèmes numériques. Marcel et al. [1997] fournissent quelques éléments sur la précision des calculs en indiquant qu'elle dépend de la résolution numérique employée lors du calcul des FFT. Foroosh et al. [2002] s'intéressent à la précision de calcul en considérant que les erreurs sont causées par le bruit dans les images, par les zones non communes entre les images, et par les effets de bords et de sous échantillonnage créés par la FFT. Ils ne fournissent néanmoins pas de solutions à l'estimation de l'incertitude sur les mesures.

L'amélioration du procédé de détermination de la transformation rigide  $[R, T]$  passe pour beaucoup par la mise en œuvre de techniques élaborées qui vont améliorer la précision lors de la transformation cartésien-polaire et par l'utilisation d'algorithmes sub-pixeliques [Averbuch et al., 2003 ; Keller et al., 2005]. Dans tous ces cas, le cadre d'utilisation se concentre sur la mise en correspondance d'images très similaires. Seules les erreurs de calcul liées à la méthode sont étudiées et réduites (erreurs de niveau subpixelique dans tous les cas).

Il faut attendre l'article [Pfungsthorn et al., 2010] pour voir un auteur s'intéresser à l'estimation de l'incertitude sur le résultat en fonction des données entrantes. En effet, dans le cadre du sonar ou du radar, les caractéristiques intrinsèques des capteurs font apparaître des imprécisions sur les mesures bien supérieures à celles engendrées par la méthode mathématique mise en œuvre. De plus, le cadre réel d'utilisation ne fournit pas forcément des images (radar ou sonar) optimales pour une mise en correspondance, d'où la nécessité d'estimer, suite à la détermination d'une transformation rigide entre deux images, les incertitudes sur ce résultat. Pfungsthorn et al. [2010] insistent sur le fait que les ambiguïtés dans les données à mettre en correspondance, les problèmes de zones non communes, le bruit, sont des phénomènes qui vont engendrer une multitude de pics de corrélation. Sous réserve de normalisation, la zone entourant le pic maximum peut être considérée comme une distribution unimodale qui peut être utilisée pour estimer les incertitudes.

### 3.2 Principe

Les principes mis en œuvre pour la recherche des incertitudes sont issus de l'observation des résultats de la corrélation de phase par la *Transformée de Fourier-Mellin*. Si par exemple on observe la figure 4.25 qui représente deux signaux identiques décalés  $X_2(x) = X_1(x - \Delta x)$ , ainsi que le résultat de la corrélation de phase. On s'aperçoit que le maximum de la corrélation représente la valeur du décalage  $\Delta x$ .

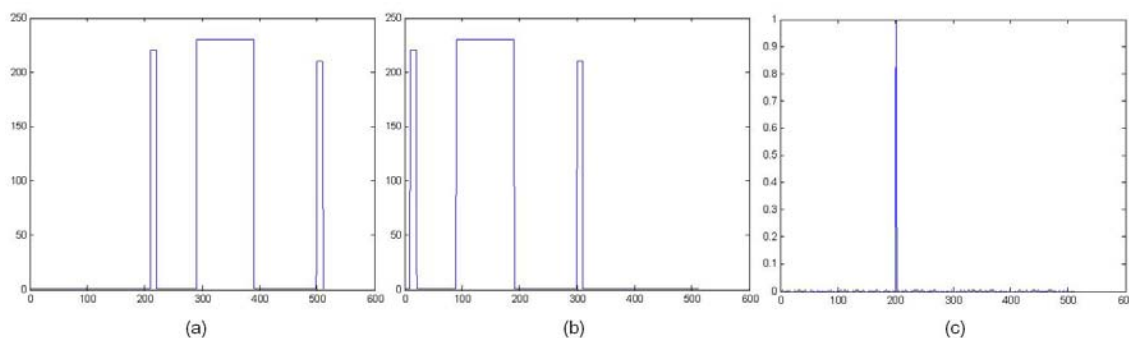


FIGURE 4.25 – Si l'on recherche la corrélation de phase (c) entre le signal  $X_1$  (a) et le signal  $X_2$  (b), on obtient un dirac positionné sur la valeur du décalage (200).

Si maintenant le signal  $X_1$  est légèrement plus large que  $X_2$ , une superposition exacte des deux signaux devient impossible, on a donc une incertitude sur le déplacement à effectuer. Le résultat de la corrélation n'est plus un dirac mais une distribution unimodale symétrique (cf. figure 4.26).

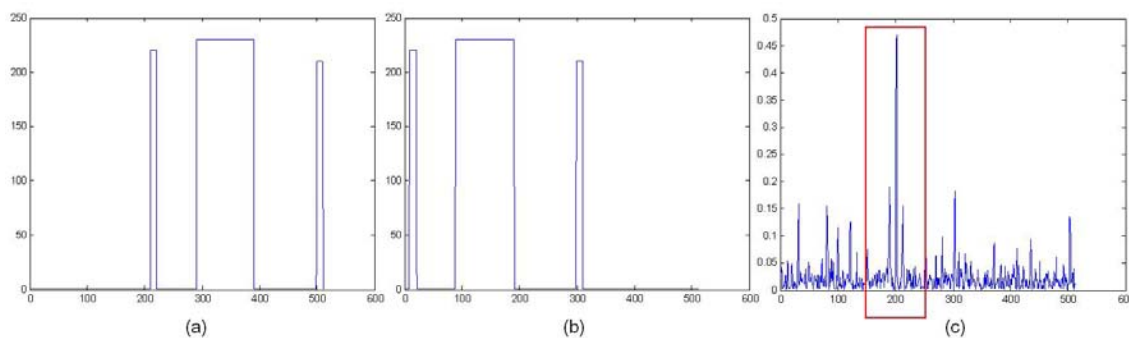


FIGURE 4.26 –  $X_1$  (a) est légèrement plus large que  $X_2$  (b), la corrélation de phase (c) donne une distribution unimodale qui montre l'incertitude sur le résultat (200). Le carré rouge montre le pic maximum pris en compte.

On note également la présence de pics locaux perturbateurs. On peut alors penser que si la recherche de la dispersion sur la distribution peut donner une idée sur l'incertitude du résultat, elle doit se faire sur la zone qui entoure le pic maximum et non sur tout le signal. Ce qui signifie également que ce procédé suppose que la recherche de la variance se fait sur une zone dont on est certain qu'elle contient bien la bonne valeur de déplacement.

Si l'on poursuit l'expérience en augmentant la disparité en largeur (cf. figure

4.27), le signal de corrélation donne une dispersion plus importante dans la zone du pic maximum.

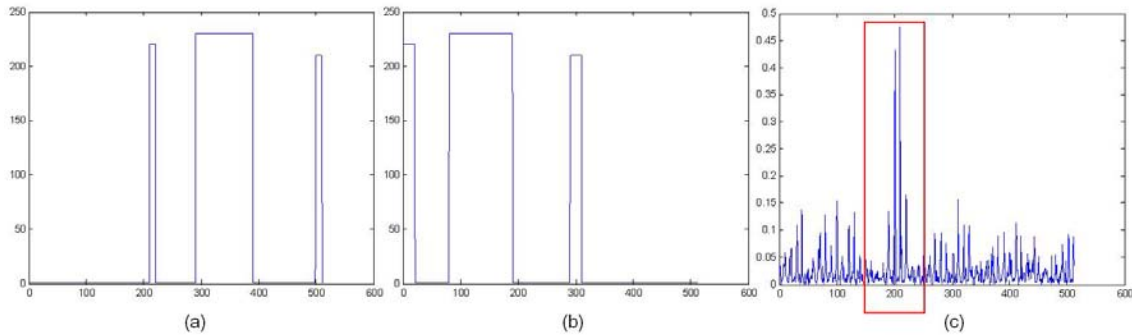


FIGURE 4.27 –  $X_1$  (a) est beaucoup plus large que  $X_2$  (b), la corrélation de phase (c) donne une distribution unimodale avec une variance plus importante. Le carré rouge indique la zone utilisée pour estimer la covariance.

Fort de ces observations, nous allons simuler un grand nombre de signaux radar une dimension afin de vérifier si l'on peut extraire de la distribution fournie par la corrélation de phase une information de covariance sur la précision des mises en correspondance.

### 3.3 Test sur un signal une dimension

L'approche est testée sur un cas simple. Il s'agit de signaux simulés une dimension.

Soit un signal à une dimension  $\mathbf{X}_1$  (cf. figure 4.28) qui représente un signal radar acquis à la position  $\mathbf{pos}_1$ .

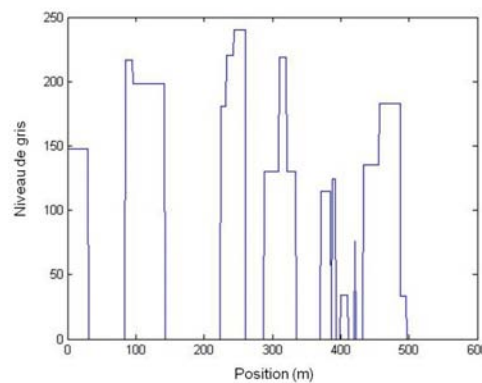


FIGURE 4.28 – Signal radar une dimension  $X_1$ . Il représente la position d'objets avec leurs niveaux de réflexions (exprimés en niveaux de gris).

Soit un signal une dimension  $\mathbf{X}_2$  (cf. figure 4.29) qui représente un signal radar, acquis dans le même environnement statique, à la position  $\mathbf{pos}_2$  distante de 20 mètres de la précédente. Ce signal est bruité par deux bruits blancs gaussiens. Le premier agit sur le niveau de réflexion des obstacles, le second sur leurs largeurs.

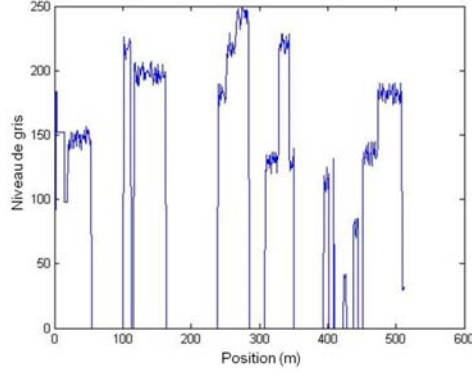


FIGURE 4.29 – Représente le signal  $X_2$ . Il est issu de  $X_1$  après un décalage de 20 mètres (avec rajout d’objets nouveaux). La largeur et le niveau de réflexion des objets sont bruités avec des bruits blancs d’écart-type :  $\sigma_{largeur} = 5\text{ m}$  et  $\sigma_{reflexion} = 10\text{ nv}$ .

Si l’on recherche la translation  $\mathbf{T}_{1,2}$  qui permet de superposer au mieux les deux signaux, on va obtenir une valeur de translation avec une incertitude d’écart-type  $\sigma_{1,2}$ . On trouve, à partir de l’équation (4.6), le signal de corrélation avec le pic indiquant la position exacte du décalage (cf. figure 4.30).

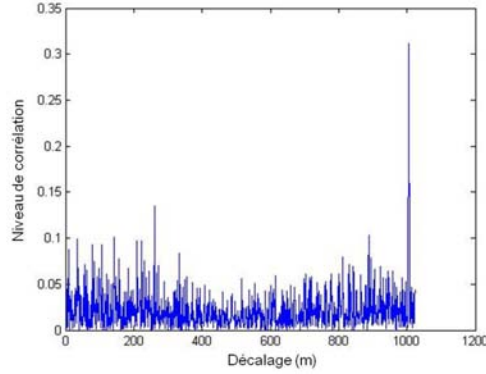


FIGURE 4.30 – Résultat de la corrélation de phase entre  $X_1$  et  $X_2$ , déterminé à partir de la méthode de la *Transformée de Fourier-Mellin*. Le pic maximum (1024-1004 = 20) représente la meilleure position estimée.

Si l’on considère que ce signal suit une distribution unimodale, on peut définir un algorithme d’estimation de la covariance (cf. algorithme 6).

Si on utilise l’algorithme 6 d’estimation de la covariance lors de la recherche de la translation entre les signaux figure 4.28 et 4.29, on obtient une covariance estimée  $Cov = \sigma^2 = (4.89)^2$ . Très proche de la valeur de bruit en translation ( $\sigma_{largeur} = 5\text{ m}$ ), cela confirme que la mise en correspondance est sensible aux variations de forme des obstacles mais très peu à leurs niveaux. Cela est normal puisque la méthode de la *Transformée de Fourier-Mellin* permet justement d’être insensible à ces variations de niveau. La figure 4.31 donne l’allure du signal de corrélation après fenêtrage.

---

**Algorithme 6:** Estimation de la covariance sur l'incertitude en translation
 

---

Calcul de la translation par la méthode de la *Transformée de Fourier-Mellin* :

$[Transl_x, Signal_{Correlation}] = ScanMatchFMT(Ref_{scan}, New_{scan})$  ;

Extraction dans  $Signal_{Correlation}$  de la zone représentant la bonne superposition :

**pour**  $i$  variant de  $Transl_x - N$  à  $Transl_x + N$  **faire**

$Signal_{Fen}(i + 1 - Transl_x + N) = Signal_{Correlation}(i)$  ;

Normalisation de  $Signal_{Fen}$  tel que :

$\sum_{i=1}^{2.N+1} Signal_{Fen} = 1$  ;

Estimation de la covariance :

$Cov = 0$  ;

**pour**  $i$  variant de  $-N$  à  $N$  **faire**

$Cov = Cov + Signal_{Fen}.i^2$  ;

---

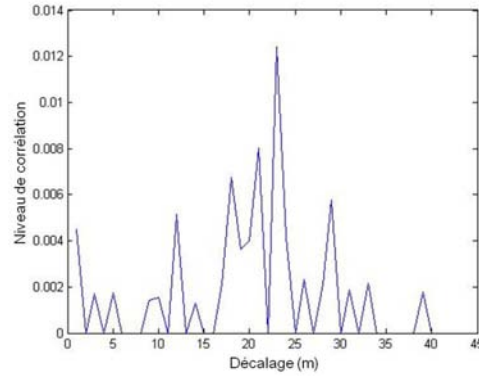


FIGURE 4.31 – Fenêtrage du signal de corrélation centré sur le pic maximum (représentation avant normalisation).

### 3.3.1 Validation du procédé

Pour valider cette méthode, l'algorithme 7 a été utilisé. Il permet de calculer, sur 800 essais, les valeurs des covariances estimées par l'algorithme 6 et de les comparer aux erreurs réelles. Ainsi, on va pouvoir déterminer de façon expérimentale les hypothèses qui ont été faites dans la section 3.2.

---

**Algorithme 7:** Validation du procédé d'estimation de la covariance
 

---

Création d'un signal de test  $X_1$  ;  
 Création d'un signal de test  $X_2$  décalé de  $\Delta x$  par rapport à  $X_1$  ;  
 $\sigma_{largeur} = 1$  ;  
 $\sigma_{reflexion} = 10$  ;  
**pour**  $i$  variant de 1 à 800 **faire**  
     **si** ( $\text{modulo}(i, 100) == 0$ ) **ET** ( $i \leq 400$ ) **alors**  
          $\sigma_{largeur} = \sigma_{largeur} + 1$  ;  
     **si** ( $\text{modulo}(i, 100) == 0$ ) **ET** ( $i > 400$ ) **alors**  
          $\sigma_{largeur} = \sigma_{largeur} - 1$  ;  
     Bruitage du signal  $X_2$  ;  
     Calcul de  $[Transl_x(i), Signal_{Correlation}] = ScanMatchFMT(X_1, X_2)$  ;  
     Calcul de  $Cov(i) = EstimCov(Transl_x, Signal_{Correlation})$  ;  
 Tracé de la courbe  $Transl_x$  qui regroupe toutes les translations déterminées ( $\Delta x$  étant la bonne valeur).  
 Tracé des courbes  $3 \times \sqrt{Cov}$  et  $-3 \times \sqrt{Cov}$  qui représentent les limites hautes et basses de l'erreur en fonction des covariances calculées.

---

La figure 4.32 représente les translations déterminées par la *Transformée de Fourier-Mellin* et les limites fixées par l'estimation des covariances.

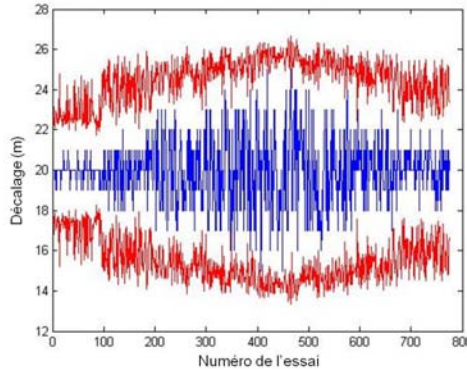


FIGURE 4.32 – Translations déterminées par la méthode de la *Transformée de Fourier-Mellin*. 800 essais bruités, pour une valeur de translation correcte de 20 m. Les courbes en rouge représentent les valeurs limites fixées par l'estimation de la covariance ( $\pm 3 \times \sqrt{Cov}$ ).

On voit donc que la covariance estimée donne une bonne estimation sur la qualité du calcul de la translation.

### 3.3.2 Conclusion

L'observation du signal de corrélation de phase entre deux signaux très similaires et décalés ressemble à une distribution gaussienne. Pour autant, est-il possible d'en déduire la matrice de covariance associée au déplacement estimé ?

Après un test expérimental sur 800 essais, la réponse est affirmative. On obtient bien, une estimation de l'incertitude sur la mise en correspondance de signaux bruités.

La section suivante présente cette technique sur des données bi-dimensionnelles simulées.

### 3.4 Test sur un signal deux dimensions

En utilisant les algorithmes 6 et 7, on teste le procédé sur deux panoramiques simulés (cf. figure 4.33) constitués d'obstacles de largeurs et de niveaux de réflexion variables. La deuxième image est issue de la première après un décalage (en mètre)  $(\Delta x, \Delta y) = (20, 30)$  et un bruit sur la largeur des obstacles d'écart-type (en mètre)  $(\sigma x, \sigma y) = (2, 5, 2, 5)$ .

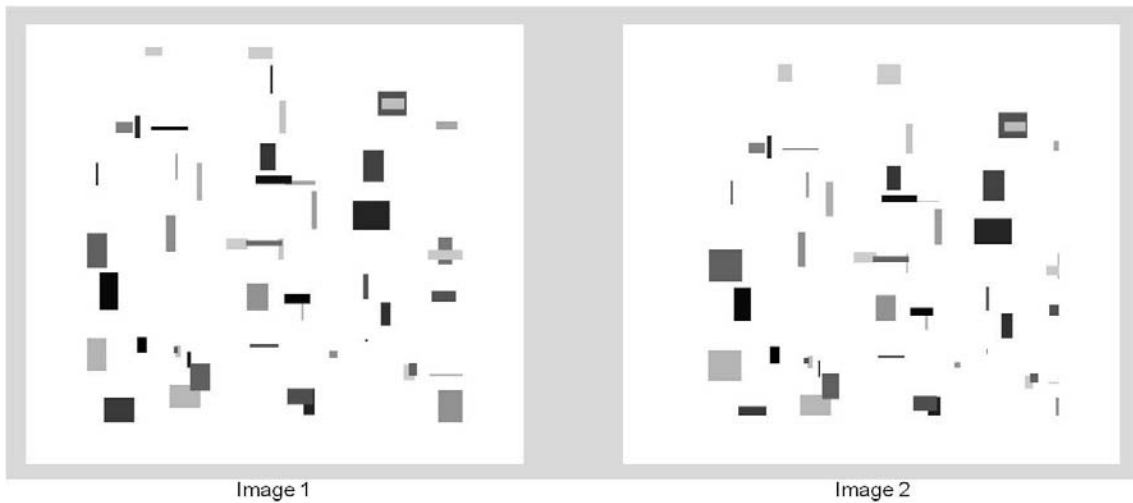


FIGURE 4.33 – Deux panoramiques simulées. L'image 1 est celle de référence. L'image 2 est issue d'une translation (en mètre) de  $(\Delta x, \Delta y) = (20, 30)$  et d'un bruitage en taille des obstacles avec des écarts types (en mètre)  $(\sigma x, \sigma y) = (2, 5, 2, 5)$ .

La zone d'intérêt utilisée pour calculer les covariances est représentée figure 4.34. Elle est issue de l'image de corrélation fournie par l'algorithme de la *Transformée de Fourier-Mellin*. Elle donne un  $(\sigma x_{estimate}, \sigma y_{estimate}) = (2, 67, 2, 29)$  proche des valeurs réelles.

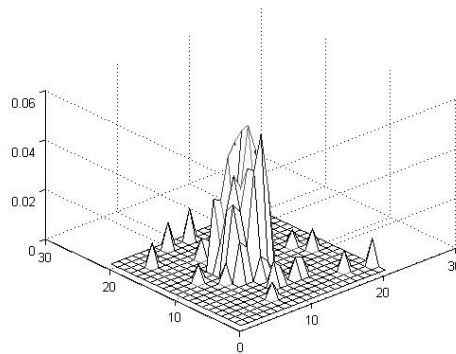


FIGURE 4.34 – Zone d'intérêt, extraite de l'image de corrélation, qui représente les probabilités en  $x$  et  $y$  de bonne localisation.

La figure 4.35 montre les résultats sur 80 tests successifs avec une variation de taille des obstacles réalisée par un bruit blanc avec un écart-type variant de  $0,5\text{ m}$  à  $4,5\text{ m}$ .

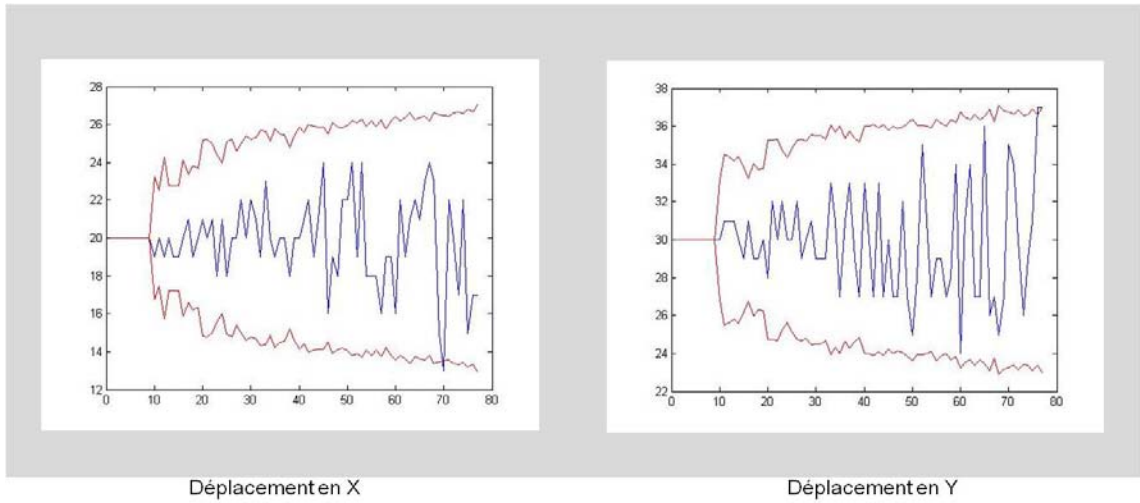


FIGURE 4.35 – L'image 1 et l'image 2 sont décalées d'une translation (en mètre)  $(\Delta x, \Delta y) = (20, 30)$ . L'ajout d'erreurs sur la taille des obstacles provoque des erreurs lors de la recherche de cette translation. En rouge les limites de ces variations calculées par l'algorithme d'estimation des covariances  $(\pm 3 \times \sqrt{Cov})$ .

En conclusion, le procédé fonctionne de façon analogue à celui utilisé sur les signaux une dimension.

Il reste maintenant à valider cette méthode sur des panoramiques réels issus d'acquisitions radar.

### 3.5 Données réelles

Réaliser un test avec des données réelles est une chose délicate. En effet, on doit disposer d'informations « vraies » pour pouvoir réaliser un calcul d'erreurs. Cette vérité est fournie, dans notre cas, par un GPS-RTK qui donne un positionnement centimétrique lorsqu'il capte de façon optimale les satellites. Dans le cas contraire cela peut rapidement décroître vers une précision métrique. Il faut enfin signaler que le GPS ne fournit pas d'information de cap, ce dernier est donc calculé à partir des positions GPS.

Le test présenté ici va permettre la validation de l'estimation de la covariance. Il a été réalisé sur des données issues d'un SLAM radar. La figure 4.36 est une vue aérienne de la zone observée par le radar. On retrouve en rouge le tracé GPS du parcours réalisé.

Nous avons choisi pour chaque panoramique (96 au total sur ce parcours) de le comparer avec les panoramiques précédents distants de moins de  $5\text{ m}$  de lui. De cette façon on obtient un comparatif avec des panoramiques proches qui logiquement sont très ressemblants et donc doivent aboutir à un *scan matching* précis. Puis, de façon à tester le procédé, le parcours boucle et de grosses variations de cap très rapides sont effectuées, le *scan matching* s'effectue alors entre des images qui peuvent être très différentes, il est donc moins précis.





FIGURE 4.36 – Zone de test. En rouge la vérité terrain issue d'un GPS-RTK.

Le procédé employé pour valider l'estimation de la covariance est donné dans l'algorithme 8.

---

**Algorithme 8:** Algorithme de validation de la covariance sur données réelles.

---

**pour**  $i$  variant de 1 à Nombre de panoramique **faire**  
    Recherche des panoramiques anciens distants de moins de 5 m ;  
    Calcul des transformations rigides et covariances associées ;  
    Calcul des transformations rigides à partir des positions GPS ;  
    Calcul des erreurs entre  $Transf_{Estim}$  et  $Transf_{GPS}$  ;  
Tracé des erreurs ;  
Tracé des courbes  $3 \times \sqrt{(Covariance)}$  et  $-3 \times \sqrt{(Covariance)}$  qui  
représentent les limites hautes et basses de l'erreur en fonction des  
covariances calculées ;

---

Les résultats sont représentés sur les figures suivantes.

La figure 4.37 donne les erreurs en translation  $X$  et les covariances associées ;

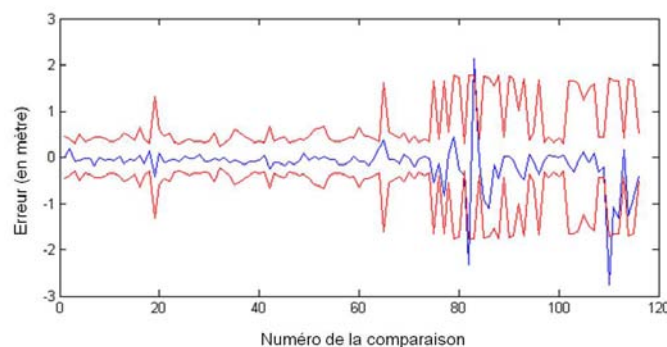


FIGURE 4.37 – Erreurs en  $X$  entre les transformations rigides calculées et les transformations rigides vraies. En rouge, les limites hautes et basses définies par les covariances calculées.

La figure 4.38 donne les erreurs en translation  $Y$  et les covariances associées.

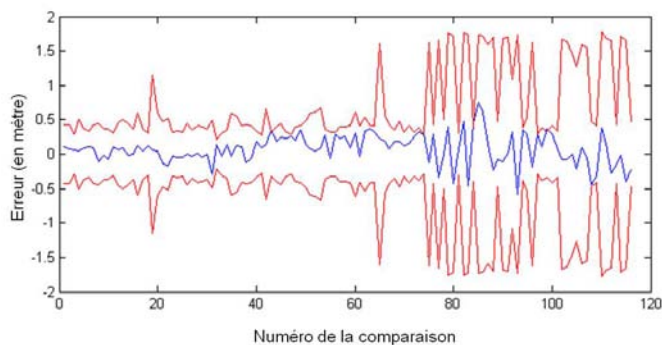


FIGURE 4.38 – Erreurs en  $Y$  entre les transformations rigides calculées et les transformations rigides vraies. En rouge, les limites hautes et basses définies par les covariances calculées.

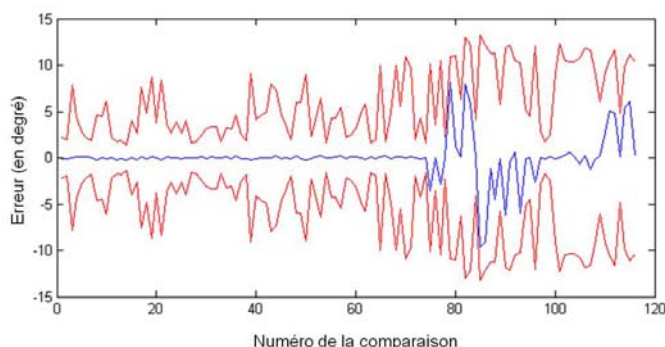


FIGURE 4.39 – Erreurs en angle entre les transformations rigides calculées et les transformations rigides vraies. En rouge les limites hautes et basses définies par les covariances calculées.

La figure 4.39 donne les erreurs en orientation et les covariances associées.

Enfin la figure 4.40 donne le parcours effectué avec la position exacte des acquisitions. Elle indique également quelques numéros de comparaisons afin de pouvoir situer les zones de comparaison.

Les comparaisons de 1 à 71 s'effectuent entre des panoramiques proches (distants de moins de 5 m et acquis avec un intervalle de temps de quelques secondes). On obtient alors des *scan matching* précis. Les covariances calculées corroborent ces résultats. Seule la comparaison 21 est moins précise car elle s'effectue entre deux panoramiques moins ressemblants. La covariance calculée l'indique en donnant des valeurs plus importantes.

Pour les comparaisons entre 72 et 115, on a une alternance de comparaisons entre panoramiques proches et panoramiques éloignés (panoramiques distants d'un point de vue temporel). De plus, des mouvements brusques (changements de caps) du véhicule mettent en défaut le modèle d'évolution du véhicule ce qui donne des panoramiques de mauvaises qualités et donc des comparaisons imprécises. Le *scan matching* fournit alors des résultats en dents de scie, résultats confirmés par l'estimation de la covariance.

Le procédé n'est pas idéal puisque dans trois cas, les covariances (81, 82, et 110) estimées fournissent des limites inférieures aux erreurs réelles (cf. figure 4.37). Ils concernent les cas limites où les transformations sont généralement exclues (car

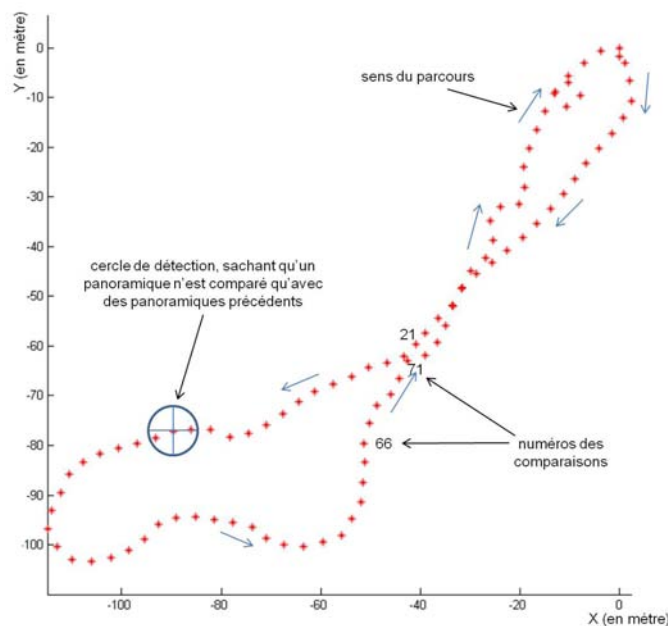


FIGURE 4.40 – Positions des acquisitions. Le cercle bleu indique les limites imposées pour une recherche de mise en correspondance.

fausses) avant même le calcul des covariances.

Enfin, si l'on souhaite parler des erreurs d'orientation, on s'aperçoit que les remarques précédentes sont valables. En revanche, il faut garder à l'esprit que le cap a été déduit des positions et donc que sa fonction de « vérité terrain » peut être remise en question.

### 3.6 Conclusion

Les résultats obtenus montrent que l'on peut estimer les incertitudes sur les transformations rigides calculées. Cela va permettre à l'algorithme EKF-SLAM de disposer d'informations extéroceptives associées à une matrice de covariance représentant de façon réaliste les bruits de mesure. On peut d'ailleurs se reporter au test réel du chapitre 5 section 3.3 qui utilise les mêmes données pour réaliser un SLAM radar. Le fait d'utiliser à ce moment là, l'estimation de covariance calculée (notamment de la comparaison au point 21) permet d'éviter au SLAM de diverger comme le fait une simple prédiction par les capteurs proprioceptifs (cf. figure 5.10).

## 4 Algorithme RS-SLAM-FMT

### 4.1 Introduction

Depuis les premières utilisations du *scan matching* dans le SLAM [Lu et Milios, 1994], la majorité des algorithmes estimatifs utilisent une méthode de résolution agissant sur la totalité des contraintes. L'algorithme EKF-SLAM est, lui, surtout mis en œuvre dans les SLAM avec carte d'amers [Leonard et Durrant-whyte, 1991a]. On retrouve néanmoins quelques rares cas d'utilisations dans le SLAM orienté trajectoire

tel que [Cole et Newman, 2006]. Ils utilisent dans ce cas un *laser SICK*, comme capteur extéroceptif.

A partir de ces études, un algorithme fondé sur l'EKF-SLAM adapté à l'utilisation du radar K2Pi est créé. Son appellation « *Radar Scan - Simultaneous Localisation And Mapping - Fourier Mellin Transform* » (RS-SLAM-FMT) résume en une simple abréviation toutes les techniques mises en œuvre.

Les sections suivantes vont rappeler le fonctionnement de l'algorithme estimatif EKF-SLAM puis décrire le *RS-SLAM-FMT*.

## 4.2 Algorithme général de l'EKF-SLAM

Le filtre de Kalman appliqué à un contexte de localisation et de cartographie simultanées a été présenté par [Dissanayake et al., 2001] sur un SLAM avec carte d'amers. Dans le schéma général du SLAM (cf. figure 4.41) il correspond au bloc d'estimation. Ce filtre est composé de quatre étapes distinctes : la prédiction (proprioception), l'observation, la mise à jour des états et la cartographie des nouveaux amers. Chacune de ces étapes (exceptée l'étape de mise à jour) est régie par des équations non-linéaires. L'objectif de ce paragraphe est de donner les équations qui régissent les différentes étapes d'un EKF-SLAM. Pour une explication plus détaillée des variables modélisant le système, on peut se reporter à l'annexe A.

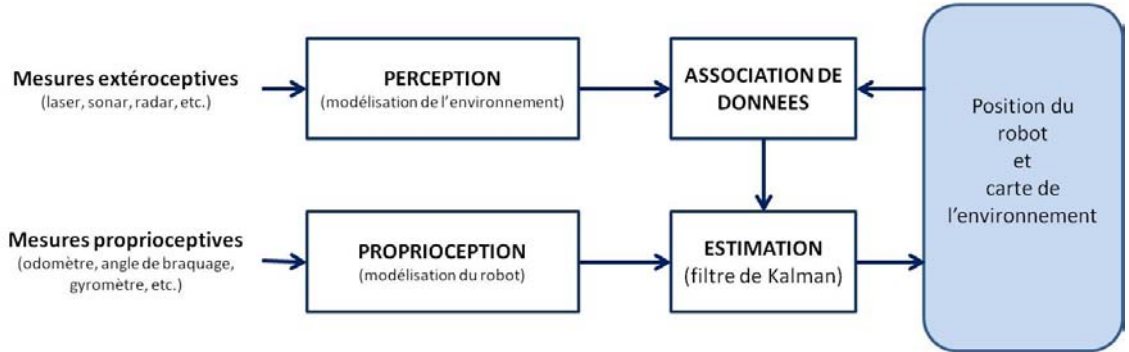


FIGURE 4.41 – Les différentes étapes de l'algorithme EKF-SLAM.

Le système est décrit par un ensemble de paramètres également appelés variables d'état. L'ensemble de ces  $n$  variables est écrit sous la forme d'un vecteur d'état :

$$\mathbf{X} = [x_1 \dots x_i \dots x_n]^T \quad (4.12)$$

Dans une application SLAM avec carte d'amers, ces paramètres  $x_i$  correspondent d'une part à la position en deux dimensions d'un véhicule et d'autre part à des amers constituant une carte de l'environnement dans lequel évolue le véhicule.

A chaque itération, trois étapes de calcul sont réalisées. Il y a tout d'abord une phase de prédiction qui s'appuie sur la fonction  $\mathbf{f}$  de l'évolution des états de  $\mathbf{X}$  :

$$\hat{\mathbf{X}}(k+1|k) = \mathbf{f}(\hat{\mathbf{X}}(k|k), \hat{\mathbf{U}}(k)) \quad (4.13)$$

$$\mathbf{P}(k+1|k) = \mathbf{F}_X \cdot \mathbf{P}(k|k) \cdot (\mathbf{F}_X)^T + \mathbf{F}_U \cdot \mathbf{Q}_U \cdot (\mathbf{F}_U)^T \quad (4.14)$$

Dans un second temps, les observations prédites (positions des objets composant l'environnement) sont calculées à l'aide de la fonction  $\mathbf{h}$  :

$$\hat{\mathbf{Z}}(k+1|k) = \mathbf{h}(\hat{\mathbf{X}}(k+1|k)) \quad (4.15)$$

Après la mise en correspondance des mesures et des éléments de la carte, une innovation est calculée. Cette innovation  $\nu$  correspond à la différence entre les positions estimées et mesurées d'une même entité de la carte. Elle est associée à une matrice de covariance  $\mathbf{S}$ .

$$\nu(k+1) = \mathbf{Z}_{\text{mesure}}(k+1) - \hat{\mathbf{Z}}(k+1|k) \quad (4.16)$$

$$\mathbf{S} = \mathbf{H}_X \cdot \mathbf{P}(k+1|k) \cdot (\mathbf{H}_X)^T + \mathbf{R} \quad (4.17)$$

Le vecteur  $\mathbf{Z}_{\text{mesure}}(k+1)$  correspond aux mesures extéroceptives associées à une matrice de covariance  $\mathbf{R}$ .

Au final, une correction de l'estimation de  $\hat{\mathbf{X}}$  est réalisée à partir de l'innovation  $\nu$ . Cette correction est pondérée par un coefficient  $\mathbf{W}$  qui permet de prendre en compte les incertitudes sur les variables de  $\hat{\mathbf{X}}$  et sur les mesures  $\mathbf{Z}$ .

$$\hat{\mathbf{X}}(k+1|k+1) = \hat{\mathbf{X}}(k+1|k) + \mathbf{W} \cdot \nu(k+1) \quad (4.18)$$

$$\mathbf{P}(k+1|k+1) = \mathbf{P}(k+1|k) - \mathbf{W} \cdot \mathbf{S} \cdot \mathbf{W}^T \quad (4.19)$$

Le gain du filtre de Kalman,  $\mathbf{W}$ , se calcule avec l'équation :

$$\mathbf{W} = \mathbf{P}(k+1|k) \cdot (\mathbf{H}_X)^T \cdot \mathbf{S}^{-1} \quad (4.20)$$

Le récapitulatif des différentes équations du filtre de Kalman non-linéaire clôt la description de l'algorithme EKF-SLAM. La suite de ce document se concentre sur la description de l'algorithme RS-SLAM-FMT qui n'utilise pas une carte d'amers mais un SLAM orienté trajectoire avec *scan matching*. L'adaptation de l'algorithme estimatif prend également en compte les particularités du radar K2Pi et du véhicule de test utilisé.

### 4.3 RS-SLAM-FMT

Le système est un SLAM orienté trajectoire. En effet, l'extraction d'amers au sein de panoramiques est délicate alors que l'on remarque que deux panoramiques proches sont très similaires. Pour cette raison, une technique de recherche d'appariements entre les vues est possible.

Le *scan matching* va déterminer les translation et rotation à mettre en œuvre sur le dernier panoramique acquis afin d'obtenir un maximum de correspondance avec les autres panoramiques. Ces transformations vont ensuite être utilisées comme nouvelles observations dans l'algorithme de SLAM et comme mise à jour des estimations lors de la détection de boucles (la trajectoire du véhicule se croise).

Le vecteur d'état à l'instant  $k$  est constitué des positions successives du véhicule :

$$\hat{\mathbf{X}}(k) = [\mathbf{x}_{v_1}^T, \dots, \mathbf{x}_{v_{k-1}}^T, \mathbf{x}_{v_k}^T]^T \quad (4.21)$$

$\mathbf{x}_{v_i} = [x_i, y_i, \phi_i]^T$  représente la position et l'orientation du véhicule à l'instant  $i$ . Chaque vecteur est associé à un panoramique radar représentant l'environnement du véhicule à cette position.

Au départ, le vecteur d'état contient la position initiale du véhicule, et la matrice de covariance  $\mathbf{P}$  une matrice  $3 \times 3$  nulle.

#### 4.3.1 Prédiction

Étant donné un vecteur de commande bruité  $\hat{\mathbf{U}}$ ,  $\mathbf{u}(k+1)$  correspond à la commande imposée pour passer de la position à l'instant  $k$  à la position à l'instant  $k+1$ . La position du véhicule peut être prédite par :

$$\mathbf{x}_{v_{n+1}}(k+1|k) = f(\mathbf{x}_{v_n}(k|k), \mathbf{u}(k+1)) = \mathbf{x}_{v_n}(k|k) \oplus \mathbf{u}(k+1) \quad (4.22)$$

$\mathbf{x}_{v_{n+1}}(k+1|k)$  est l'estimé de  $\mathbf{x}_{v_{n+1}}$  à la  $(n+1)$  position du véhicule à l'instant  $(k+1)$ .  $\oplus$  représente l'opérateur de composition défini dans [Smith et al., 1990].

Dans notre cas, la fonction  $\mathbf{f}(\hat{\mathbf{X}}, \hat{\mathbf{U}})$  représente la loi d'évolution du véhicule en fonction des données de contrôle.

Les données de contrôle sont issues de deux capteurs proprioceptifs : un odomètre et un gyromètre. Ils permettent de fournir la distance parcourue et le changement d'orientation du véhicule entre deux positions successives.

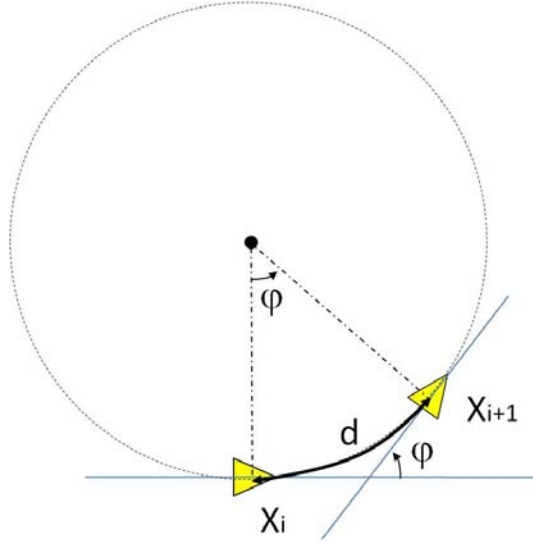


FIGURE 4.42 – Modèle d'évolution du véhicule (modèle point).

Le véhicule expérimental utilisé fournit des données odométriques et gyrométriques, et des panoramiques radars, tous centrés sur son essieu avant. Le déplacement du véhicule entre deux enregistrements va être approximé à un cercle (voir figure 4.42). On peut écrire les équations suivantes :

$$\mathbf{u}(k+1) = \begin{pmatrix} \mathbf{d} \times \frac{\sin(\varphi)}{\varphi} \\ \mathbf{d} \times \frac{(1-\cos(\varphi))}{\varphi} \\ -\varphi \end{pmatrix} \text{ si } \varphi \neq 0 \quad (4.23)$$

$$\mathbf{u}(k+1) = \begin{pmatrix} \mathbf{d} \\ 0 \\ 0 \end{pmatrix} \text{ si } \varphi = 0 \quad (4.24)$$

- $d$  est la distance parcourue par le véhicule entre l'instant  $k$  et l'instant  $k+1$  (fournie par l'odomètre),
- $\varphi$  est le changement d'orientation du véhicule entre l'instant  $k$  et l'instant  $k+1$  (fourni par le gyromètre).

Si on définit  $\mathbf{P}_{vn}(k|k)$  la matrice de covariance qui représente les incertitudes sur la position du véhicule à l'instant  $k$ , et si on note  $\mathbf{Q}_U$  la matrice de covariance qui représente l'incertitude sur les bruits du système de contrôle, alors on peut écrire l'équation (4.25)<sup>1</sup>.

$$\mathbf{P}_{v(n+1)}(k+1|k) = \mathbf{J}_1(\mathbf{x}_{vn}, \mathbf{u})\mathbf{P}_{vn}(k|k)\mathbf{J}_1(\mathbf{x}_{vn}, \mathbf{u})^T + \mathbf{J}_2(\mathbf{x}_{vn}, \mathbf{u})\mathbf{Q}_U\mathbf{J}_2(\mathbf{x}_{vn}, \mathbf{u})^T \quad (4.25)$$

$\mathbf{J}_1$  et  $\mathbf{J}_2$  sont les jacobienues de l'opérateur de composition ( $\oplus$ ).

$$\mathbf{J}_1(\mathbf{x}_1, \mathbf{x}_2) = \frac{\delta(\mathbf{x}_1 \oplus \mathbf{x}_2)}{\delta \mathbf{x}_1} \quad (4.26)$$

$$\mathbf{J}_2(\mathbf{x}_1, \mathbf{x}_2) = \frac{\delta(\mathbf{x}_1 \oplus \mathbf{x}_2)}{\delta \mathbf{x}_2} \quad (4.27)$$

Le vecteur d'état global  $\hat{\mathbf{X}}$  est constitué des positions successives du véhicule. Il va donc être augmenté avec la nouvelle position prédite  $\mathbf{x}_{v_{n+1}}(k+1|k)$ . Sa matrice de covariance  $\mathbf{P}$  va également être augmentée :

$$\hat{\mathbf{X}}(k+1|k) = \begin{bmatrix} \hat{\mathbf{X}}(k|k) \\ \mathbf{x}_{v_n} \oplus \mathbf{u}(k+1) \end{bmatrix} \quad (4.28)$$

$$\mathbf{P}(k+1|k) = \begin{bmatrix} \mathbf{P}(k|k) & \mathbf{P}(k|k)\mathbf{J}_1(\mathbf{x}_{vn}(k|k), \mathbf{u})^T \\ \mathbf{J}_1(\mathbf{x}_{vn}(k|k), \mathbf{u})\mathbf{P}(k|k)^T & \mathbf{P}_{v(n+1)}(k+1|k) \end{bmatrix} \quad (4.29)$$

### 4.3.2 Mise à jour

#### Fonction de cartographie

La carte de localisation est constituée de toutes les positions estimées du véhicule. A chaque position est associée un panoramique. La figure 4.43 représente les différentes positions du véhicule et les zones observées. La figure 4.44 donne l'exemple de trois panoramiques associés à trois positions estimées.

Lors de la recherche de  $\hat{\mathbf{X}}(k+1|k+1)$ , on dispose du panoramique enregistré à l'instant  $k+1$ , mais aussi des panoramiques enregistrés aux instants précédents 0 à  $k$ .

Le vecteur des mesures extéroceptives  $\mathbf{Z}_{mesure}$  est constitué des transformations rigides  $[R, T]$  qui représentent le déplacement du véhicule entre la position  $\mathbf{x}_v(k+1)$  et les positions précédentes. Pour cela, on utilise l'algorithme de la *Transformée de Fourier-Mellin* pour le calcul des correspondances entre panoramiques.

---

1. avec les informations  $k$  et  $k+1$  supprimées de  $\mathbf{x}_{vn}$  et  $\mathbf{u}$  afin de ne pas surcharger la formule.

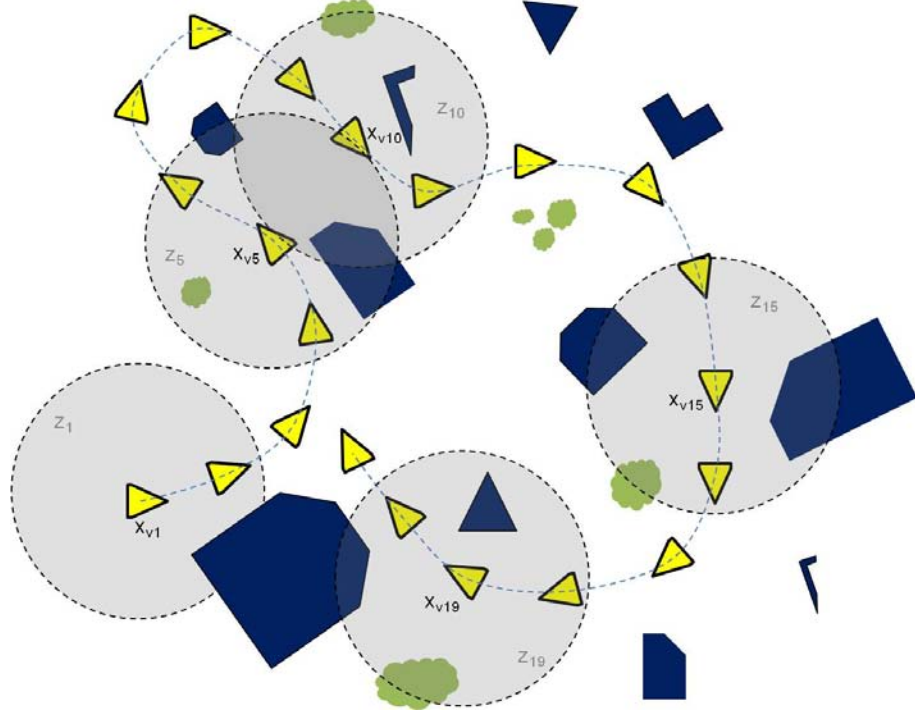


FIGURE 4.43 – SLAM dans un environnement péri-urbain. Les limites d'acquisition du capteur extéroceptif (radar) sont représentées par un cercle (seul quelques panoramiques sont affichés).

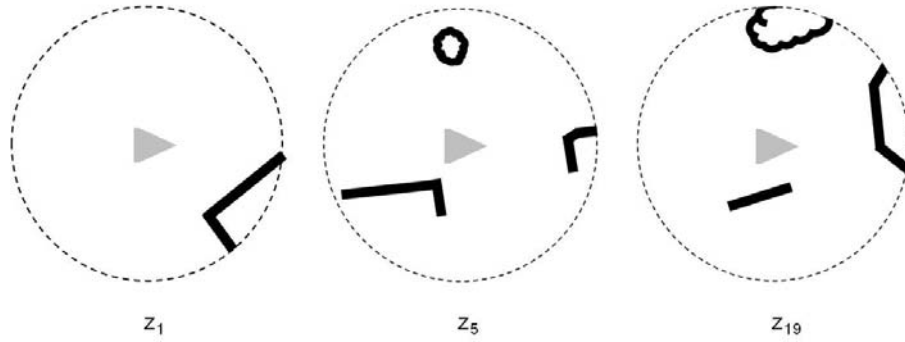


FIGURE 4.44 – Exemples de panoramiques radar pour les positions  $x_{v1}, x_{v5}, x_{v10}$ .

### Mise à jour

Pour l'étape de mise à jour, il est nécessaire de calculer l'innovation  $\nu$  (équation (4.35)). Elle est fonction des observations réelles  $\mathbf{Z}_{mesure}(k+1)$  et des observations estimées  $\hat{\mathbf{Z}}(k+1|k)$ .

Si on définit :

$$\mathbf{rt}_{i,j} = [(\Delta x_{i,j}, \Delta y_{i,j}), \theta_{i,j}] \quad (4.30)$$

$\mathbf{rt}_{i,j}$  représente la transformation rigide déterminée entre le panoramique  $Pano_i$  acquis à la position  $\mathbf{x}_{vi}$  et le panoramique  $Pano_j$  acquis à la position  $\mathbf{x}_{vj}$ .

Dans l'exemple de la figure 4.45, on peut déterminer :

- les observations estimées.



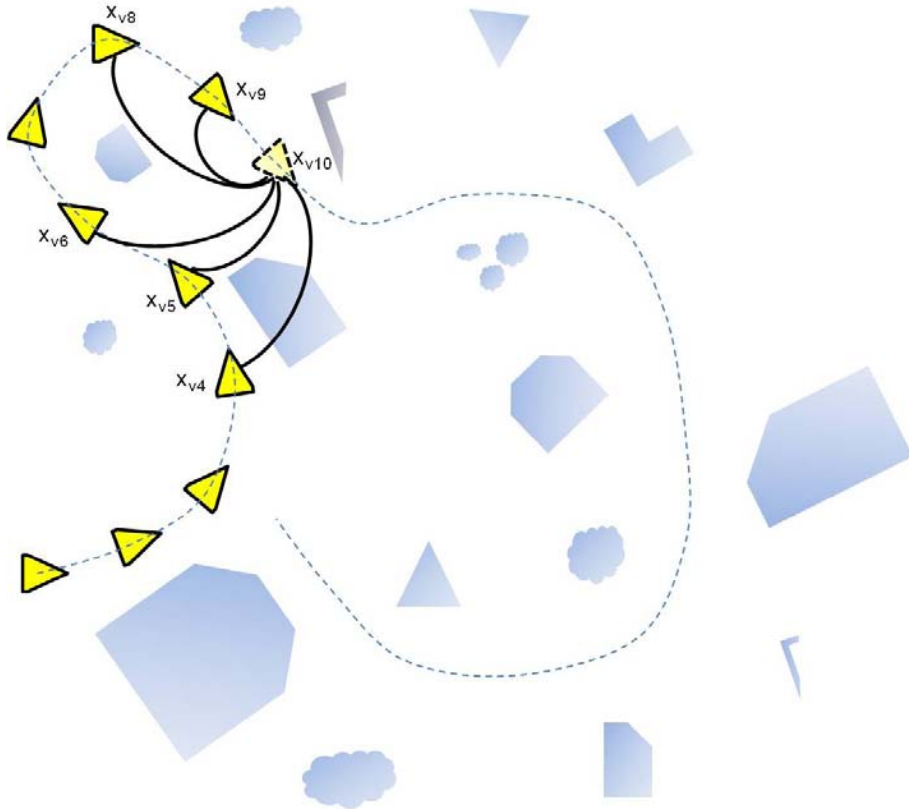


FIGURE 4.45 – Mises en correspondances utilisées pour la mise à jour de la position  $x_{v10}$ .

$\hat{\mathbf{Z}}(10|9)$  a pour expression :

$$\hat{\mathbf{Z}}(10|9) = \begin{bmatrix} \mathbf{z}_{10,4} \\ \mathbf{z}_{10,5} \\ \mathbf{z}_{10,6} \\ \mathbf{z}_{10,8} \\ \mathbf{z}_{10,9} \end{bmatrix} \quad (4.31)$$

avec :

$$\mathbf{z}_{i,j}(k) = h(\mathbf{x}(k+1|k)) = (\ominus \mathbf{x}_{vj}(k+1|k) \oplus \mathbf{x}_{vi}(k+1|k)) \quad (4.32)$$

– les observations mesurées. Elles sont déterminées par la *Transformée de Fourier-Mellin*.

$$\mathbf{Z}_{mesure}(10) = \begin{bmatrix} \mathbf{rt}_{10,4} \\ \mathbf{rt}_{10,5} \\ \mathbf{rt}_{10,6} \\ \mathbf{rt}_{10,8} \\ \mathbf{rt}_{10,9} \end{bmatrix} \quad (4.33)$$

Le modèle d'observation est du type :

$$\mathbf{Z}(k) = \mathbf{h}(\mathbf{X}(k)) + \mathbf{r}(k) \quad (4.34)$$

avec  $\mathbf{r}()$  : bruit de mesure

Comme [Bailey et Durrant-Whyte, 2006 ; Cole et Newman, 2006], on peut écrire les équations suivantes :

$$\nu = \mathbf{Z}_{mesure}(k+1) - \hat{\mathbf{Z}}(k+1|k) \quad (4.35)$$

$$\mathbf{S} = \nabla \mathbf{H}_{\mathbf{x}} \cdot \mathbf{P}(k+1|k) \cdot \nabla \mathbf{H}_{\mathbf{x}}^T + \mathbf{R}(k+1) \quad (4.36)$$

$$\mathbf{W} = \mathbf{P}(k+1|k) \cdot \nabla \mathbf{H}^T \cdot \mathbf{S}^{-1} \quad (4.37)$$

$\mathbf{R}$  représente la matrice de covariance associée au bruit de mesure sur les observations (elle est issue de l'algorithme de mise en correspondance de *Fourier Mellin*).

On voit que seul  $\nabla \mathbf{H}_{\mathbf{x}}$  n'est pas encore déterminée. Si on utilise  $\mathbf{J}_1$  et  $\mathbf{J}_2$  définis dans les équations (4.26)(4.27), on peut déterminer.

$$\nabla \mathbf{H}_{\mathbf{x}} = \frac{\delta \mathbf{h}}{\delta \mathbf{x}} \begin{bmatrix} \frac{\delta \mathbf{h}}{\delta \mathbf{x}}_{q=0} \\ \frac{\delta \mathbf{h}}{\delta \mathbf{x}}_{q=1} \\ \vdots \\ \frac{\delta \mathbf{h}}{\delta \mathbf{x}}_{q=n} \end{bmatrix} \quad (4.38)$$

avec :

$$\frac{\delta \mathbf{h}}{\delta \mathbf{x}_q} = \begin{cases} 0 & si \ q \neq i, j \\ \mathbf{J}_1(\ominus \mathbf{x}_{vi}(k+1|k), \mathbf{x}_{vj}(k+1|k)) \times \mathbf{J}_{\ominus}(\mathbf{x}_{vi}(k+1|k)) & si \ q = i \\ \mathbf{J}_2(\ominus \mathbf{x}_{vi}(k+1|k), \mathbf{x}_{vj}(k+1|k)) & si \ q = j \end{cases} \quad (4.39)$$

avec  $\mathbf{J}_{\ominus}$  la jacobienne de la transformée inverse :

$$\mathbf{J}_{\ominus}(\mathbf{x}_1) = \frac{\delta(\ominus \mathbf{x}_1)}{\delta(\mathbf{x}_1)} \quad (4.40)$$

Il ne reste plus qu'à effectuer la mise à jour  $x(k+1|k+1)$  et celle de la matrice de covariance associée  $P(k+1|k+1)$ .

$$\hat{\mathbf{X}}(k+1|k+1) = \hat{\mathbf{X}}(k+1|k) + \mathbf{W}\nu \quad (4.41)$$

$$\mathbf{P}(k+1|k+1) = \mathbf{P}(k+1|k) - \mathbf{W} \cdot \mathbf{S} \cdot \mathbf{W}^T \quad (4.42)$$

### 4.3.3 Bouclage

Dans tout problème de SLAM, l'incertitude sur la position du robot croît au fil du temps. Si on considère que les capteurs (odomètre, gyromètre, radar) sont bruités (bruit de type gaussien), il apparaît évident que les positions du robot, calculées à partir de ces capteurs, possèdent une incertitude qui augmente à chaque utilisation des données capteur (cf. figure 4.52).

Si une boucle apparaît, c'est-à-dire si le véhicule croise dans son parcours une position déjà rencontrée, il peut, via le filtre de Kalman corriger son incertitude (cf. figure 4.53).

On s'aperçoit que la difficulté du bouclage se situe au niveau de la détection même de cette boucle. Dans un SLAM à base d'amers, cela va nécessiter des algorithmes d'association de données complexes afin de ne pas effectuer de mauvais appariements.

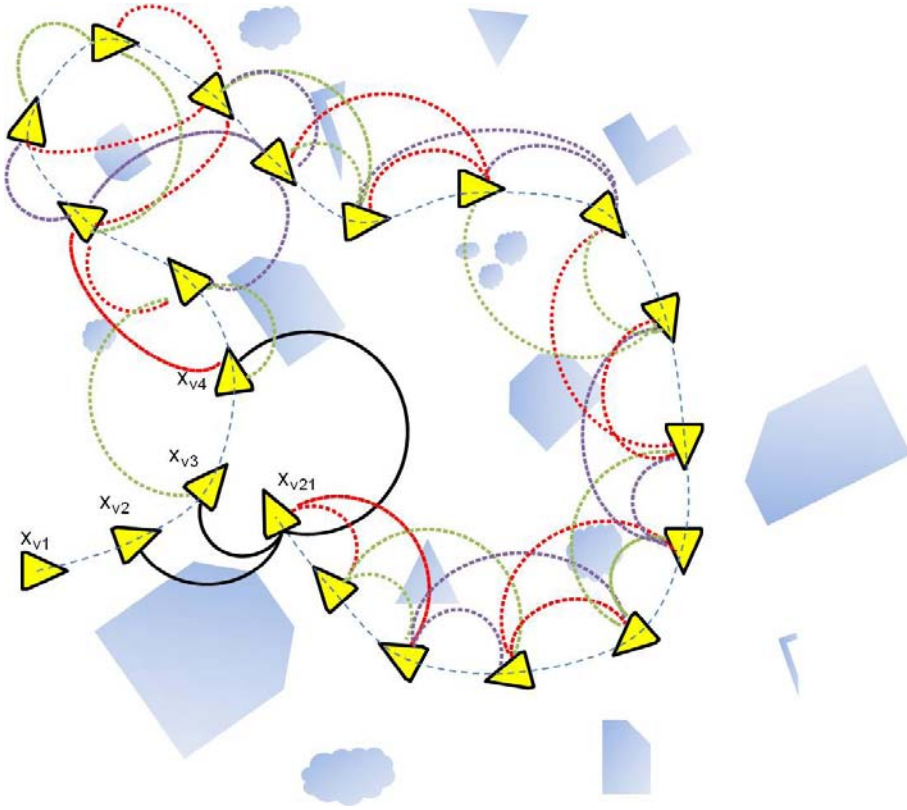


FIGURE 4.46 – Bouclage : le bouclage associe  $x_{v21}$  avec  $x_{v2}, x_{v3}, x_{v4}$ . Les liens en pointillés montrent les interactions entre les mises à jours des  $x_{vi}$ . En conclusion, ce bouclage va influencer les estimées de toutes les positions de  $x_{v5}$  à  $x_{v21}$ .

Le cadre de notre étude se base sur la mise en correspondance de panoramiques. Nous allons donc utiliser les positions estimées (vecteur d'état  $\hat{\mathbf{X}}$ ) et les incertitudes correspondantes (extraites de la matrice de covariance  $\mathbf{P}$ ) afin de limiter la zone de recherche des panoramiques à associer (figure 4.46).

Une fois toutes ces positions trouvées (chacune étant associée à un panoramique radar), on calcule toutes les transformations rigides entre ces panoramiques et le nouveau panoramique acquis. Il ne reste plus qu'à utiliser le coefficient de robustesse pour déterminer les meilleurs appariements (en cas de coefficients trop faibles, aucun appariement n'est jugé viable, aucune mise à jour n'est effectuée).

#### 4.3.4 Carte de représentation

La carte de représentation a pour objectif d'offrir au manipulateur du véhicule d'exploration une carte dense de l'environnement qu'il pourra utiliser pour planifier le parcours.

Si on définit  $\hat{\mathbf{X}}(k) = [\mathbf{x}_{v_k}^T, \mathbf{x}_{v_{k-1}}^T, \dots, \mathbf{x}_{v_1}^T]^T$  le vecteur d'état déterminé par le RS-SLAM-FMT, et qui donne les positions successives du robot, avec  $Pano_j$  qui est le panoramique observé à la position  $\mathbf{x}_{v_j}$ , on peut alors créer une carte globale en respectant le principe des grilles d'occupation (voir le chapitre 3 section 4.2).

L'algorithme 9 présente le principe utilisé pour la construction de la carte dense. Il indique que cette carte est la somme des panoramiques placés dans un repère

global, offrant ainsi un reflet de la présence probable d'objets dans la scène.

---

**Algorithme 9:** Construction d'une carte dense de l'environnement
 

---

```

Création d'une matrice  $Carte = [N_{lignes}, M_{colonnes}]$  nulle ;
Création d'une matrice de normalisation  $Norm = Carte$  ;
pour  $i$  variant de 1 à Nombre de positions estimées faire
    Seuillage de  $Pano_i$  pour l'élimination des niveaux inférieurs à  $Seuil_{min}$  ;
    Positionnement de  $Pano_i$  dans le repère de la carte :
     $PanoCarte_i = TransformRepereGlobal(Pano_i, \mathbf{x}_{v_i})$  ;
    Ajout du  $Pano_i$  dans la carte :
     $Carte = Carte + PanoCarte_i$  ;
    pour  $x$  variant de 1 à  $N_{lignes}$  faire
        pour  $y$  variant de 1 à  $N_{colonnes}$  faire
            si  $PanoCarte_i(x, y)$  existe alors
                 $Norm(x, y) = Norm(x, y) + 1$  ;
         $Carte = Carte ./ Norm$  ;
    
```

---

La figure 4.54 donne un exemple de reconstruction d'un environnement péri-urbain.

## 5 Validation du procédé

### 5.1 Introduction

Les essais décrits dans cette section ont été réalisés sur un site péri-urbain. Ils sont issus d'une campagne de tests menés dans le cadre de la conférence « Field and Service Robotics » [Checchin et al., 2010]. Le radar (disposé sur l'essieu avant du véhicule à une hauteur de 3 mètres) et les capteurs proprioceptifs sont montés sur un véhicule se déplaçant à une vitesse moyenne de 15 km/h.

Le radar K2Pi fournit un panoramique toutes les secondes. Les acquisitions de toutes les données sont enregistrées en temps réel. En ce qui concerne le traitement par l'algorithme RS-SLAM-FMT, celui-ci s'effectue à partir des enregistrements (post-traitement).

Un GPS RTK, monté sur le véhicule, fournit la vérité terrain.

### 5.2 Expérimentation urbaine sans bouclage

La campagne s'est déroulée sur le campus universitaire Blaise Pascal de Clermont-Ferrand. Cet environnement péri-urbain contient quelques éléments mobiles qui seront, dans cette application, considérés comme de simples perturbations.

Le campus est constitué d'un grand nombre de bâtiments, de parkings, de terrains de sports ... Cela va permettre de réaliser des observations radar denses et qui ne présentent pas de zones ambiguës.

L'objectif est pour ce premier essai, d'une part de fournir suffisamment d'informations caractéristiques à l'algorithme de *Fourier-Mellin* pour éviter les mauvais

appariements (voir la section 2.5.1), et d'autre part de permettre un test des différents algorithmes développés sous *Matlab*.

La figure 4.47 donne une image satellite de la zone couverte, avec en superposition les positions GPS du véhicule et les positions estimées par l'algorithme de *RS-SLAM-FMT*. La figure 4.48 représente seulement le parcours effectué afin de mieux visualiser la qualité de l'estimation par rapport à la vérité terrain.

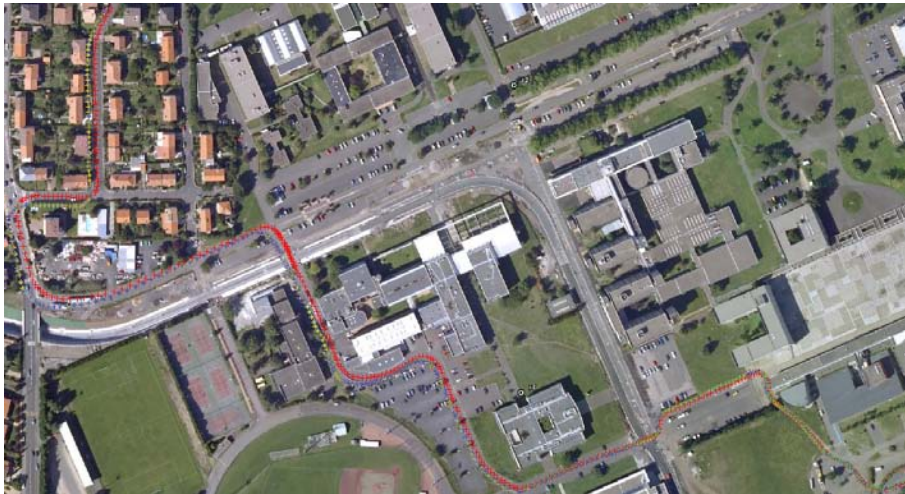


FIGURE 4.47 – Parcours de 1135 m, en rouge la vérité terrain (GPS RTK), en bleu le parcours estimé par l'algorithme *RS-SLAM-FMT*.

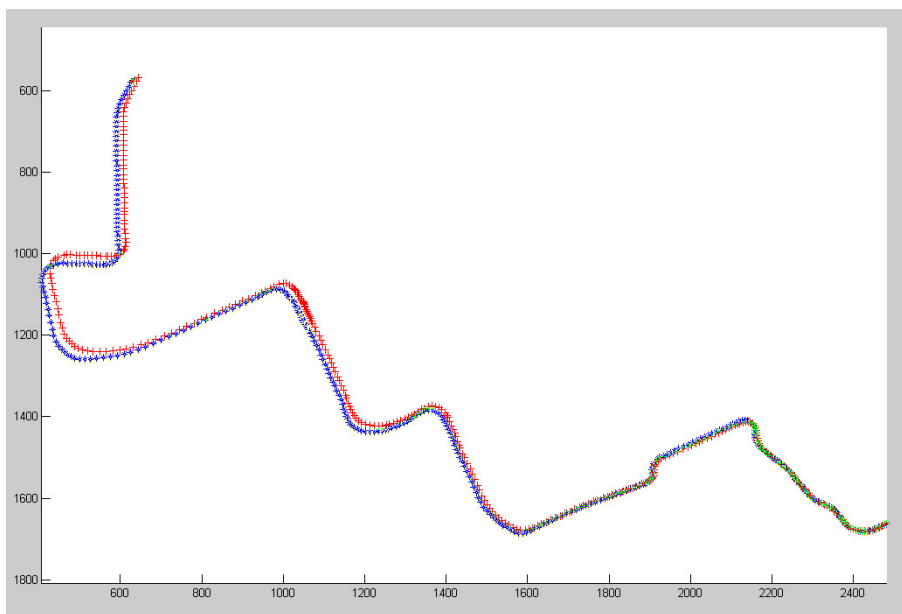


FIGURE 4.48 – En rouge la vérité terrain (GPS RTK), en bleu le parcours estimé par l'algorithme *RS-SLAM-FMT*. Le repère est gradué en mètres.

La figure 4.49 fournit la carte reconstruite à partir des positions estimées dans la figure 4.48 et des panoramiques fournis par le radar K2Pi (voir la figure 4.50).



FIGURE 4.49 – Carte reconstruite à partir des positions estimées.

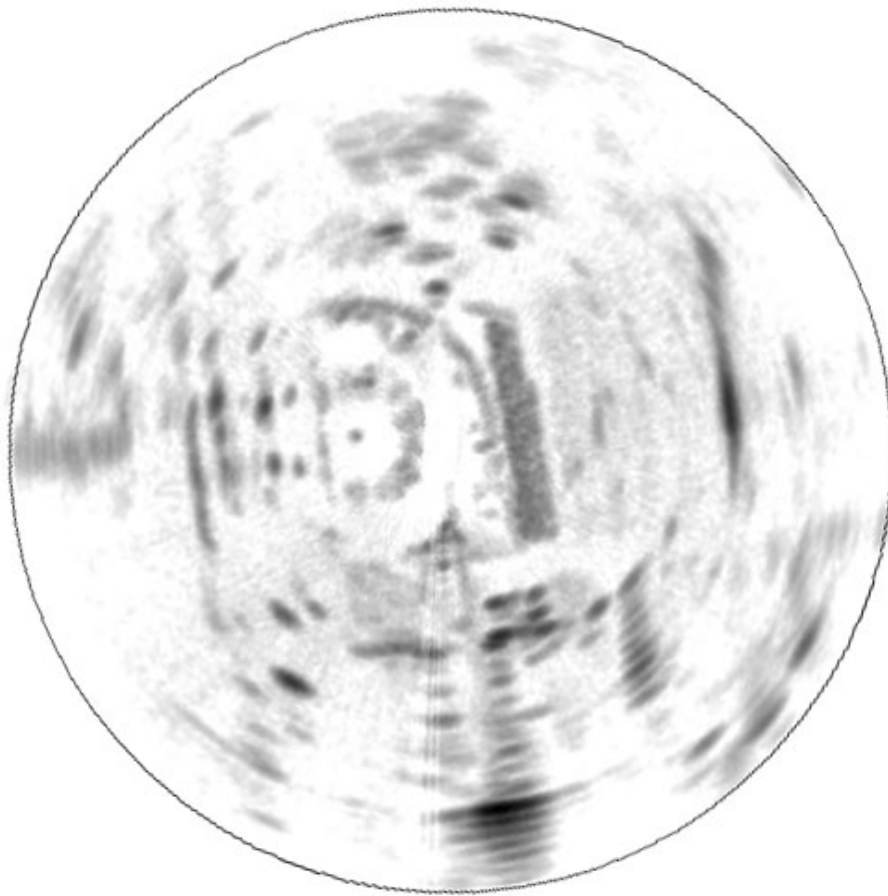


FIGURE 4.50 – Exemple d'un panoramique issu du radar K2Pi. Le cercle représente la limite de détection du radar.

### 5.3 Expérimentation urbaine avec bouclage

On réalise une nouvelle campagne. Cette fois, le véhicule va réaliser un bouclage. Il va donc revenir traverser sa trajectoire. L'objectif est de vérifier que le bouclage permet à l'algorithme d'affiner ses estimations.

La figure 4.51 donne une vue aérienne de la zone.

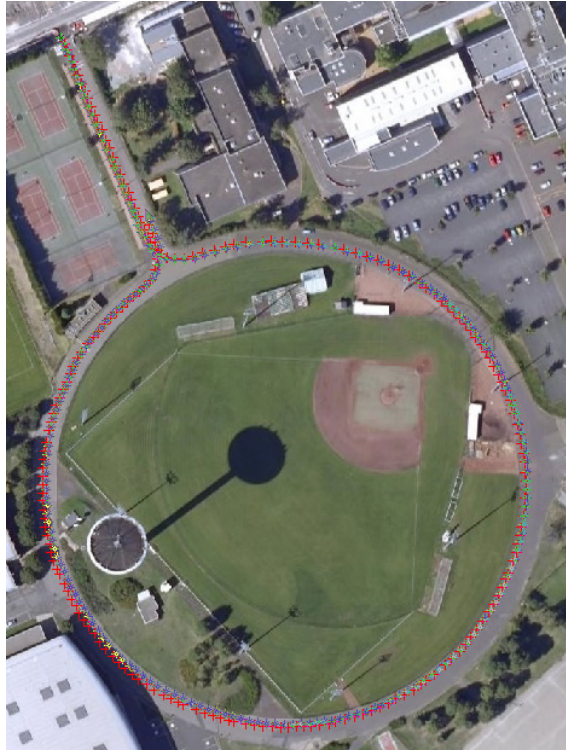


FIGURE 4.51 – Parcours de 700 m, en rouge la vérité terrain (GPS RTK), en bleu le parcours estimé par l'algorithme *RS-SLAM-FMT*.

La figure 4.52 donne le parcours estimé (en bleu) avec l'incertitude (fournie par la matrice de covariance) estimée de chaque position, juste avant le bouclage. La figure 4.53 fournit les mêmes données mais juste après le bouclage. On y voit clairement que l'estimation des incertitudes donne de meilleures valeurs, les estimations des positions sont meilleures.

La figure 4.54 représente la reconstruction de la carte à l'aide des panoramiques issus du radar K2Pi et des positions estimées.



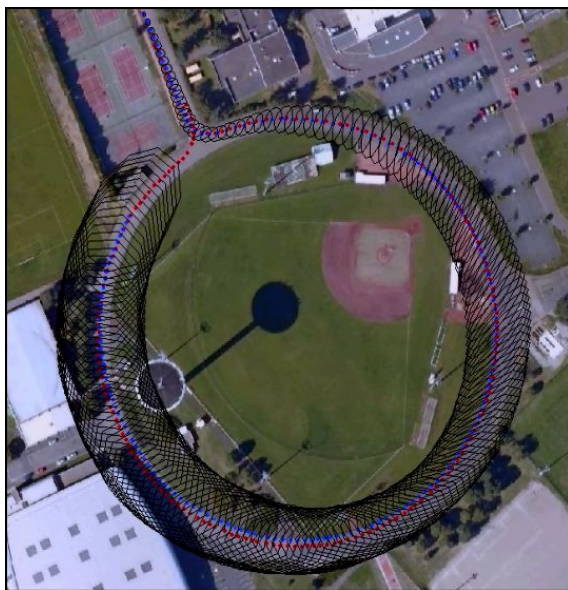


FIGURE 4.52 – Parcours estimé avec superposition des estimées des incertitudes de positionnement. Juste avant le bouclage.

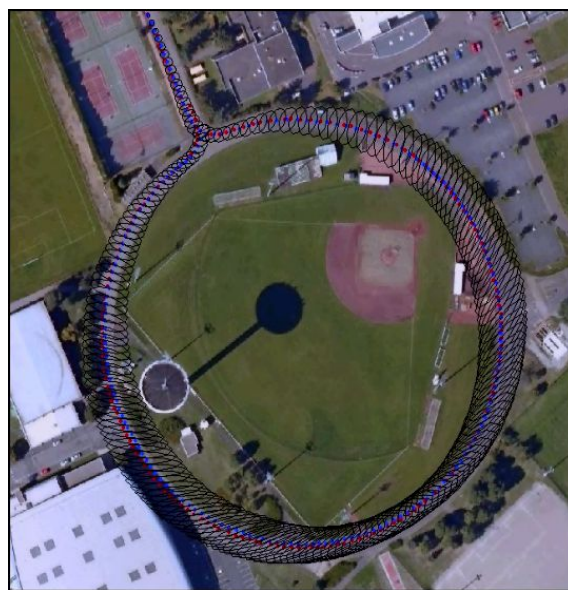


FIGURE 4.53 – Parcours estimé avec superposition des estimées des incertitudes de positionnement. Juste après le bouclage.

## 5.4 Conclusion

Ce chapitre vient de présenter le *RS-SLAM-FMT*. Associé au radar K2Pi il permet de réaliser une cartographie et une localisation simultanées dans un milieu péri-urbain.

Il utilise un *scan matching* sur des panoramiques radar novateur. Fondé sur la *Transformée de Fourier-Mellin* (FMT), il permet, sans initialisation, de calculer la transformation rigide  $RT$  entre deux panoramiques acquis à des positions différentes. Plus robuste et aussi rapide qu'un algorithme de type ICP, il est également la pièce maîtresse d'un algorithme d'estimation de l'incertitude. Innovant, il détermine en temps réel, après chaque transformation trouvée, les incertitudes sur le résultat trouvé.

Les transformations déterminées par le *scan matching* sont ensuite fournies à un algorithme de type EKF-SLAM. Ce dernier a été adapté aux particularités des données radar, et propose une méthode de détection des boucles robuste qui assure une forte amélioration des estimations.

Enfin, via le principe des grilles d'occupation, le procédé construit une carte dense de l'environnement qui peut être directement utilisable par un opérateur.

L'application du *RS-SLAM-FMT* dans des conditions péri-urbaines est donc un succès. Reste maintenant à comparer la méthode d'estimation *EKF-SLAM* à d'autres techniques existantes afin de justifier son utilisation dans notre contexte.

Le chapitre suivant va donc s'attacher à comparer le *RS-SLAM-FMT* avec d'autres algorithmes d'estimation. Il va ensuite présenter plusieurs essais dans des environnements variés. Enfin, il va donner des explications sur l'implémentation du procédé sur une plateforme de développement industrielle.



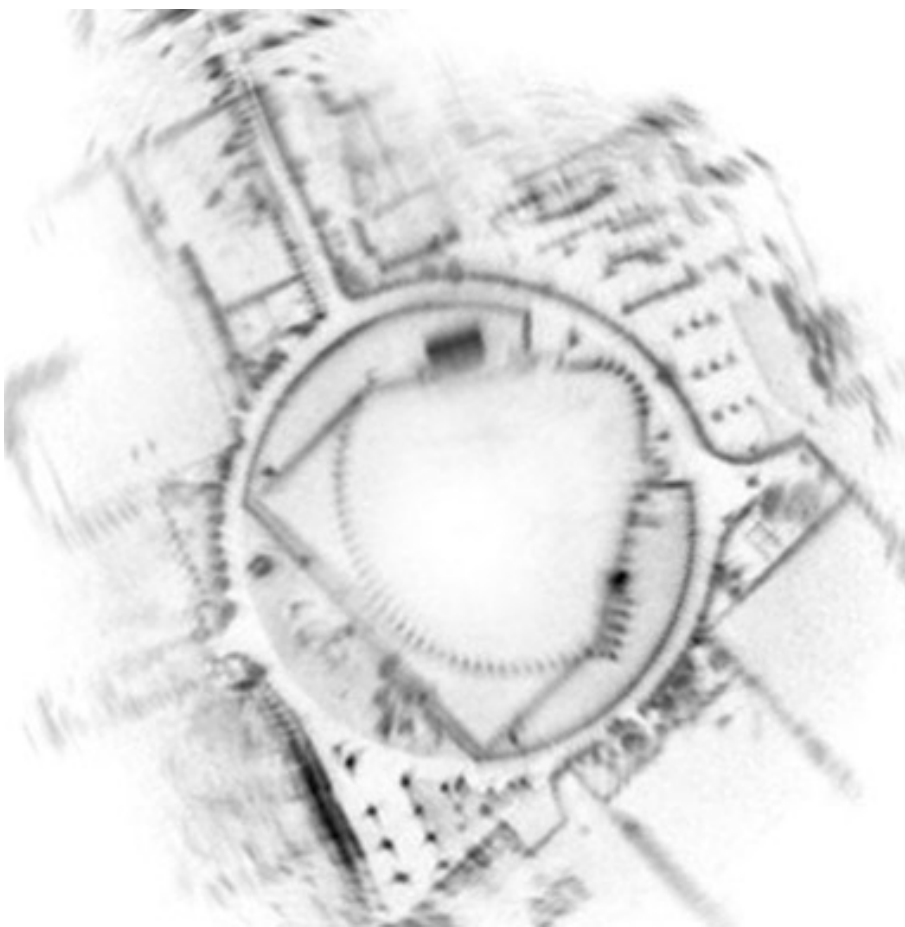


FIGURE 4.54 – Carte construite à partir des positions estimées.

# Chapitre 5

## Résultats expérimentaux et implémentation temps réel

### 1 Introduction

Le RS-SLAM-FMT a été présenté dans le chapitre précédent. Il permet, à l'aide du radar K2Pi, de réaliser une localisation et une cartographie simultanées en milieu extérieur. Les premiers tests dans un contexte péri-urbain sans la présence d'objets mobiles se sont avérés positifs. Une validation complète nécessite d'autres campagnes d'essais. Ce chapitre présente trois types d'essais qui vont permettre de finaliser la validation du procédé.

Le premier va s'attacher à réaliser un test comparatif entre le RS-SLAM-FMT et deux algorithmes plus traditionnellement utilisés dans le SLAM orienté trajectoire. Il s'agit des algorithmes d'optimisation non-linéaire *Levenberg-Marquardt* et *Square Root SAM*. L'objectif est de prouver que, dans ce SLAM orienté trajectoire, l'approche fournit des résultats équivalents ou meilleurs que ceux obtenus par ces algorithmes.

Le second essai, dans le cadre des critères imposés par le projet ANR IMPALA, va tester l'algorithme dans un environnement extérieur naturel. Les obstacles ne sont plus des immeubles mais des arbres, des bosses voire de simples buissons. Le radar est-il en mesure de fournir suffisamment de données au *scan matching* afin que ce dernier puisse calculer les transformations rigides ?

La dernière campagne d'essais teste le système lorsque le véhicule se déplace à une vitesse moyenne élevée, dans un milieu urbain chargé en objets mobiles.

Une analyse de ces essais est dressée avec une interrogation sur l'influence des capteurs proprioceptifs dans les résultats de SLAM. Cela permet d'introduire un dernier test comparatif. Il est réalisé entre le RS-SLAM-FMT et le ROLAM<sup>1</sup>. Ce procédé a également été développé dans le cadre de l'ANR IMPALA, avec pour objectif de s'affranchir des capteurs proprioceptifs. Ce n'est pas un SLAM mais un procédé qui permet d'estimer les vitesses linéaires et angulaires avec les seules données radar.

Pour conclure, le chapitre présente le dernier objectif de ces travaux qui est le développement d'un SLAM temps réel utilisant les principes qui viennent d'être décrits. A terme cela doit permettre d'implanter ce code de calcul sur une plateforme

---

1. Radar-Only Localization and Mapping

logicielle temps réel nommée « RTMaps ». Cette dernière équipe un robot autonome de la société THALES, le « R-Trooper » (cf. figure 5.4), qui devrait pouvoir, ainsi équipé, réaliser un *SLAM temps réel* dans des zones d'exploration variées.

## 2 Essais comparatifs avec d'autres algorithmes de SLAM

### 2.1 Introduction

L'objectif de cette section est d'une part de valider le *scan matching* par *Transformée de Fourier-Mellin*, par sa mise en œuvre avec des algorithmes qui sont couramment utilisés dans le SLAM orienté trajectoire (*SAM*, *MLR*,  $\sqrt{SAM}$ ), et d'autre part de comparer les résultats avec le RS-SLAM-FMT qui lui utilise l'algorithme EKF-SLAM.

La méthode par filtre de Kalman étendu souffre de ses choix de linéarisation qui entraînent une divergence du résultat. Pour résoudre ces problèmes, de nombreux chercheurs proposent l'utilisation de méthodes de minimisation globales. Elles ne se contentent pas de calculer la dernière position du véhicule, mais elles résolvent le problème d'estimer la totalité des positions connaissant les observations associées. C'est pour cette raison que ces méthodes sont souvent utilisées en fin de parcours pour déterminer la meilleure trajectoire finale.

Néanmoins, dans le cas d'une implémentation temps réel, l'EKF-SLAM permet de calculer rapidement, à chaque nouvelle observation, une nouvelle position du véhicule et la covariance associée. Un algorithme de type *SAM* optimise l'ensemble des observations et nécessite des temps de calculs plus élevés.

Ce travail de comparaison de plusieurs méthodes d'estimation pour un même type de capteur extéroceptif a été présenté lors de la conférence *IROS-2009* [Gérosier et al., 2009].

Dans cette section Les méthodes globales qui vont être utilisées en comparaison avec l'EKF-SLAM sont :

- l'algorithme de *Levenberg-Marquardt* [Frese et Hirzinger, 2001] ;
- l'algorithme *Square Root SAM* [Dellaert et Kaess, 2006].

Elles déterminent le maximum *a posteriori* (MAP) pour la trajectoire  $X$ , étant données les observations  $Z$  et les données de commande  $U$ . L'estimation du MAP est obtenue par  $X^* \triangleq \operatorname{argmax} P(X|Z)$ .

Cela revient à minimiser l'équation suivante :

$$\sum_{i=1}^N \|x_i - f_i(x_{i-1}, u_i)\|_{\Lambda_i}^2 + \sum_{k=1}^K \|z_k - h_k(x_{ik}, x_{jk})\|_{\Sigma_i}^2 \quad (5.1)$$

Avec :

- $\|e\|_{\Sigma_i}^2 \triangleq e^T \Sigma_i^{-1} e$ , la distance de Mahalanobis ;
- $x_i = f_i(x_{i-1}, u_i) + w_i$ , la prédiction de la position en fonction des données odométriques (dans notre cas, les capteurs proprioceptifs : odomètre et gyromètre), où  $w_i$  est le bruit gaussien sur les mesures, associé à la matrice de covariance  $\Lambda_i$  ;

- $z_k = h_k(x_{ik}, x_{jk}) + v_k$ , l'équation d'observation, où  $v_k$  est le bruit gaussien sur les observations, associé à la matrice de covariance  $\Sigma_i$ .

On peut se reporter aux références [Frese, 2007 ; Dellaert et Kaess, 2006] si l'on souhaite plus de détails mathématiques concernant ces deux méthodes.

## 2.2 Résultats

Pour réaliser cette comparaison, les données utilisées sont celles de la section 5.2 du chapitre 4, il n'y a donc pas de bouclage. Pour la comparaison avec *Levenberg Marquardt*, les résultats fournis représentent les erreurs d'estimations déterminées en fin de parcours car cet algorithme réalise une minimisation globale de toutes les erreurs (figure 5.2).

Pour l'algorithme *SAM*, on peut déterminer une estimation des positions du véhicule à chaque itération (une itération correspond au rajout dans le système d'une nouvelle observation) (figure 5.3).

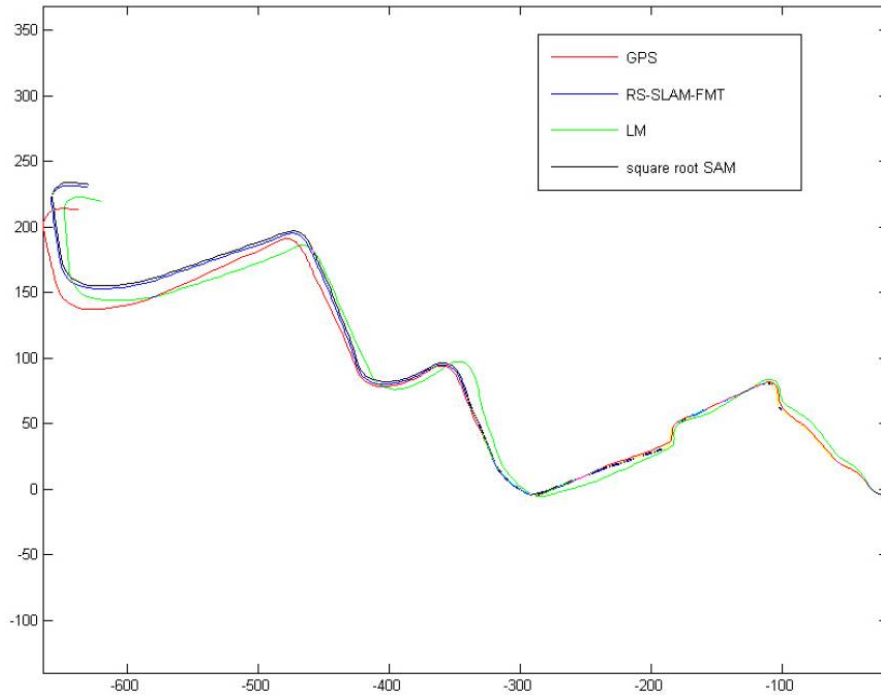


FIGURE 5.1 – Trajectoires estimées par les trois techniques. En rouge, la vérité terrain.

La figure 5.1 représente les positions estimées par les trois techniques. Les méthodes *SAM* et RS-SLAM-FMT fournissent des estimations similaires, alors que *Levenberg Marquardt* donne des résultats de plus mauvaise qualité.

La trajectoire est constituée de 290 observations, chacune est associée à une position du véhicule. La figure 5.2 donne les erreurs entre les positions estimées et la vérité terrain fournie par un GPS RTK. On voit que RS-SLAM-FMT donne le meilleur résultat. *SAM* est très proche alors que *Levenberg Marquardt* reste très en retrait en terme de qualité. Cela s'explique par le fait qu'il ne prend pas en compte

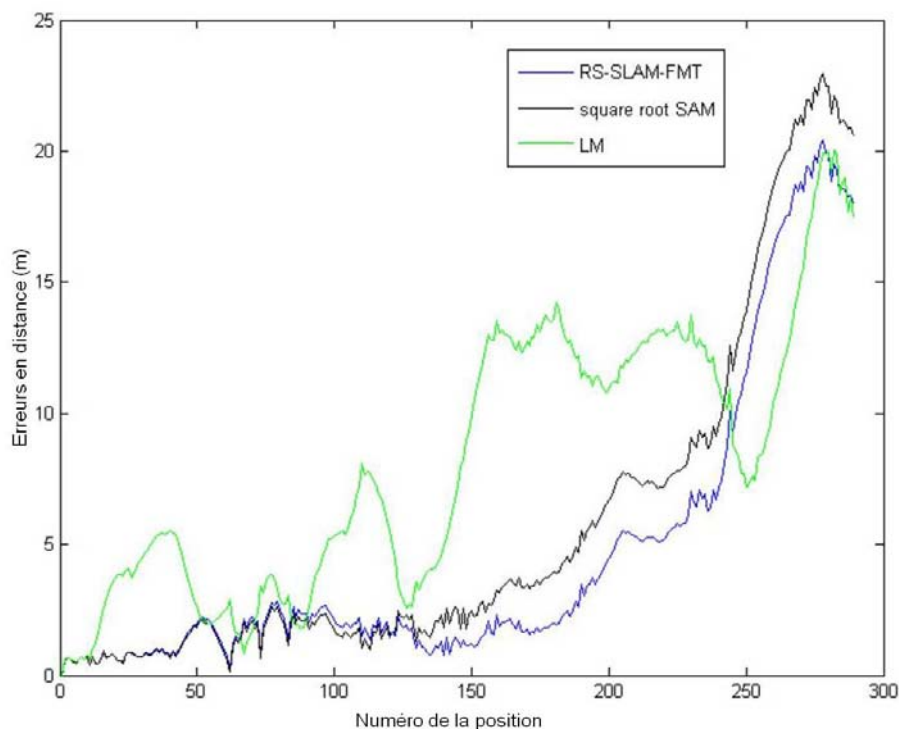


FIGURE 5.2 – Valeurs des erreurs entre les positions estimées et la vérité terrain en fin de traitement.

la matrice de covariance sur le bruit des mesures, il cherche donc à minimiser au mieux les erreurs sans prendre en compte les incertitudes sur les mesures.

Enfin, la figure 5.3 donne, à l'arrivée de chaque nouvelle observation, l'erreur moyenne entre les positions estimées et la vérité terrain. Cette information est importante car elle montre l'aptitude d'un algorithme à fournir des résultats d'estimation des positions tout au long de l'exploration : « on explore une zone inconnue, l'algorithme fournit de façon évolutive le parcours. On peut donc, à chaque instant, choisir les orientations futures ». Ce test n'a pu être réalisé qu'entre les algorithmes *SAM* et RS-SLAM-FMT. Il montre qu'un système fondé sur l'EKF-SLAM permet une estimation optimale à chaque instant, ce qui est plus difficile pour le *SAM*.

## 2.3 Conclusion

Cette partie n'avait pas pour objectif de tester la qualité des observations fournies par l'algorithme FMT, mais celui de démontrer la validité d'un algorithme EKF SLAM comme algorithme estimatif.

Les résultats présentés dans les figures 5.2 et 5.3 montrent que l'utilisation de méthodes globales ne fournit pas de meilleurs résultats pour effectuer l'estimation des positions en fonction des observations acquises. En fin de parcours, les observations sont fortement bruitées, l'utilisation de l'algorithme *SAM* qui minimise l'ensemble des erreurs, n'apporte pas de meilleurs résultats que l'algorithme RS-SLAM-FMT.

De plus, les méthodes globales apportent moins de flexibilité et, avec des algorithmes non optimisés, des temps de calcul bien supérieurs à ceux observés avec un algorithme RS-SLAM-FMT.

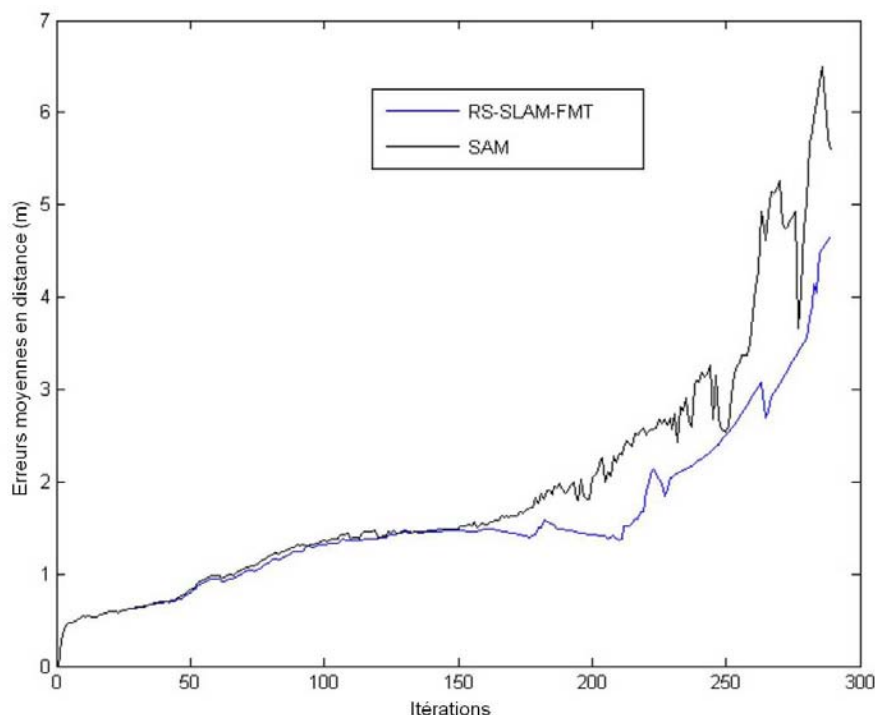


FIGURE 5.3 – A chaque itération on calcule la moyenne des erreurs entre les estimations et la vérité terrain (GPS RTK).

La section suivante présente un test de robustesse de l'algorithme lors de l'exploration d'un environnement naturel. Les zones remarquables sont rares et ne possèdent pas de formes géométriques simples.

### 3 Essais en environnement naturel

#### 3.1 Introduction

Cette partie s'appuie sur les expérimentations réalisées dans le cadre de l'ANR IMPALA. Elles furent menées avec l'objectif de tester la robustesse du procédé.

Ces essais montrent les aptitudes du couple « radar K2Pi, algorithme RS-SLAM-FMT » à réaliser une localisation et cartographie simultanées de bonnes qualités dans un environnement précis. Pour effectuer ces essais, la société THALES avait mis à disposition un véhicule d'exploration (cf. figure 5.4) équipé : d'un logiciel d'enregistrement, d'un GPS RTK (qui fournit la vérité terrain), d'un laser SICK, et bien entendu du radar K2Pi. Un SLAM SICK a été effectué en post-traitement afin de fournir un comparatif visuel avec les résultats obtenus par le RS-SLAM-FMT.

On peut retrouver en annexe B la description de tous les essais effectués pour mettre à l'épreuve l'algorithme. Cette section va présenter deux manipulations, la première est une trajectoire simple avec rebouclage long et le second une trajectoire avec des mouvements brusques et rapides.

Les deux essais présentés vont permettre de vérifier le bon fonctionnement du RS-SLAM-FMT dans les environnements et conditions suivants :

- environnement à forte densité, constitué d'arbres, de buissons, et de bâtiments



FIGURE 5.4 – R-Trooper : véhicule d’exploration de la société THALES équipé du radar K2Pi.

- à fort pouvoir de réflexion (saturation des signaux) ;
- environnement naturel à très faible densité d’objets ;
- mouvement rectiligne dans un environnement de type couloir ;
- mouvements rapides ;
- bouclage de parcours ;

### 3.2 Cas simple avec bouclage long

La figure 5.5 montre une vue aérienne de la zone couverte, avec le chemin suivi par le véhicule (vérité terrain par GPS-RTK). Il est annoté afin d’expliquer géographiquement ses différentes particularités.



FIGURE 5.5 – Vue aérienne. Le parcours du véhicule est indiqué en rouge.

Les caractéristiques de l’essai sont résumées dans le tableau 5.1. Elles indiquent notamment que le système réalise la sommation de 3 panoramiques pour créer un panoramique filtré (filtrage temporel pour l’élimination du speckle). La zone de détection du radar est un cercle de 100 m de rayon, et la vitesse moyenne du véhicule est de 5 km/h. On choisit également, de façon à limiter la taille du vecteur d’état, de ne prendre en compte un panoramique que lorsqu’il est distant de plus de 6 m par rapport au précédent.

TABLE 5.1 – Caractéristiques de l’essai de bouclage long

caractéristiques radar	
Nombre de panoramiques	149
Temps de cycle	1 s
Nombre de panoramiques sommés pour réduction du speckle	3
Déconvolution	OUI
caractéristiques du parcours	
Distance	1024 m
Bouclage	OUI
Pente	OUI
Zones denses	OUI
Zones de faible densité	OUI
caractéristiques RS-SLAM-FMT	
Distance minimum entre panoramiques	6 m
Vitesse moyenne	5 km/h
Gestion des boucles	OUI

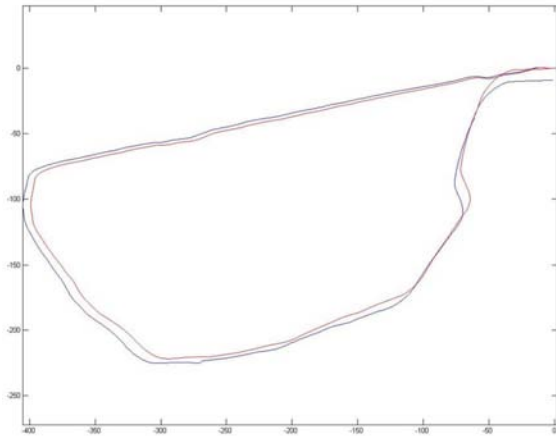
La figure 5.6 présente en premier lieu l’estimation du parcours par l’utilisation seule des données proprioceptives (odomètre et gyromètre). Ensuite un comparatif visuel entre les SLAM RS-SLAM-FMT et SLAM SICK est donné. Enfin, une représentation, sous la forme d’ellipse, des incertitudes sur les mesures estimées via le RS-SLAM-FMT (avant et après bouclage) est faite. Le bouclage du parcours apporte des informations d’observation qui permettent d’améliorer l’estimation et par conséquent la diminution des incertitudes sur les mesures.

La figure 5.7 montre sous la forme d’un graphique, les erreurs après bouclage entre la vérité terrain et le SLAM RS-SLAM-FMT. L’erreur maximale est  $E_{max} = 3,4\text{ m}$ , l’erreur moyenne est  $E_{moy} = 1,3\text{ m}$  avec un écart type  $\sigma_{erreur} = 0,8\text{ m}$ . Aucun comparatif des erreurs du SLAM SICK n’est réalisé car ce dernier n’a, dans ce travail, qu’un rôle de validation visuelle de la consistance des résultats.

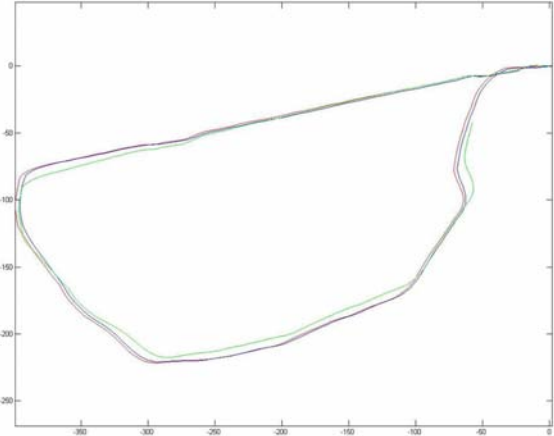
Enfin, la figure 5.8 représente une cartographie dense de l’environnement réalisée à partir des panoramiques acquis, des positions estimées par le RS-SLAM-FMT et de la méthode des grilles d’occupation. La finesse des détails, contours des bâtiments, lampadaires, arbustes et buissons sont suffisants pour apporter à un opérateur des informations sur le contenu de la zone explorée.

En conclusion, cet essai montre que l’algorithme RS-SLAM-FMT fournit des résultats en concordance avec ceux d’autres algorithmes de SLAM et d’autres capteurs extéroceptifs (ici un LIDAR). De plus, les pièges naturels de la zone tels que les saturations du capteur liées à des bâtiments en tôle, les zones désertes, les zones ambiguës et les forts dénivelés n’ont pas réussi à mettre en échec la méthode. Il faut

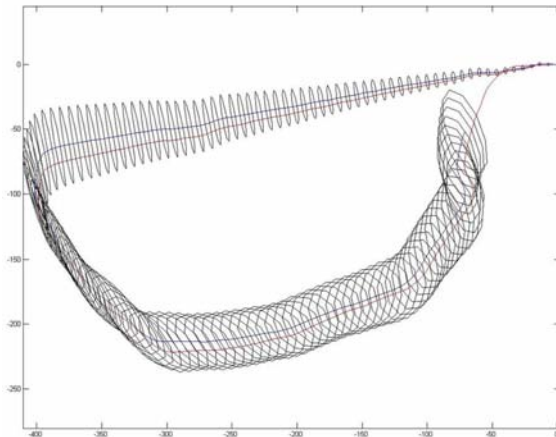




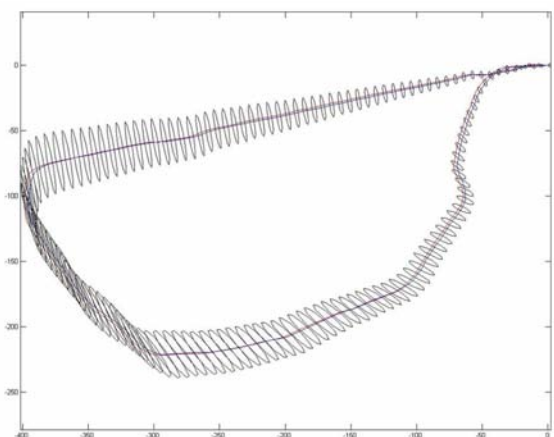
En rouge la vérité terrain (GPS RTK), en bleu l'estimation du parcours par une simple prédiction fournie par les capteurs proprioceptifs.



En rouge la vérité terrain, en bleu le RS-SLAM-FMT, en vert le SLAM SICK.



En rouge la vérité terrain, en noir les ellipses représentant les incertitudes sur les mesures. Le bouclage du parcours n'a pas encore eu lieu.



En rouge la vérité terrain, en noir les ellipses représentant les incertitudes sur les mesures. Le bouclage a eu lieu.

FIGURE 5.6 – Tracés comparatifs du parcours « bouclage long »

néanmoins rappeler que tout ceci a été effectué à une vitesse très faible (de l'ordre de 5 km/h) et sans objets mobiles présents dans la scène.

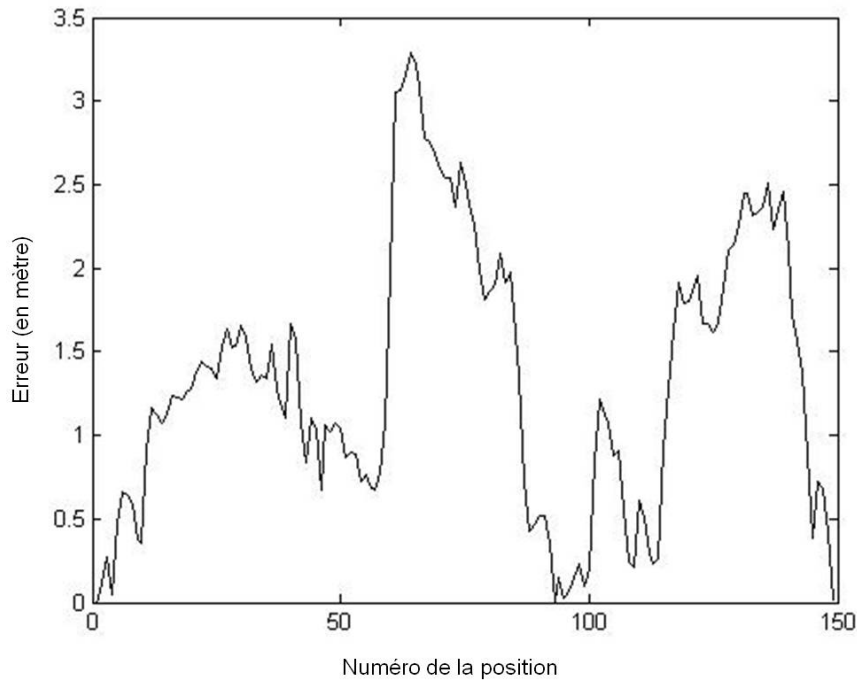


FIGURE 5.7 – Erreurs entre la vérité terrain et le parcours estimé par le SLAM RS-SLAM-FMT.

### 3.3 Trajectoire avec mouvements brusques et rapides

La figure 5.9 montre une vue aérienne de la zone couverte et du parcours suivi. Il est annoté afin d'expliquer géographiquement ses différentes particularités.

Les caractéristiques de l'essai sont résumées dans le tableau 5.2.

TABLE 5.2 – Caractéristiques de l'essai « mouvements brusques »

caractéristiques radar	
Nombre de panoramiques	96
Temps de cycle	1 s
Nombre de panoramiques sommés pour réduction du speckle	3
Déconvolution	OUI
caractéristiques du parcours	
Distance	380 m
Bouclage	OUI
Pente	NON
Zones denses	NON
Zones de faible densité	OUI
Changements rapides d'orientation	OUI
caractéristiques RS-SLAM-FMT	
Distance minimum entre panoramiques	3 m
Vitesse moyenne	5 km/h
Gestion des boucles	OUI

La figure 5.10 présente l'estimation du parcours par l'utilisation seule des données proprioceptives (odomètre et gyromètre). La figure 5.11 montre un comparatif visuel

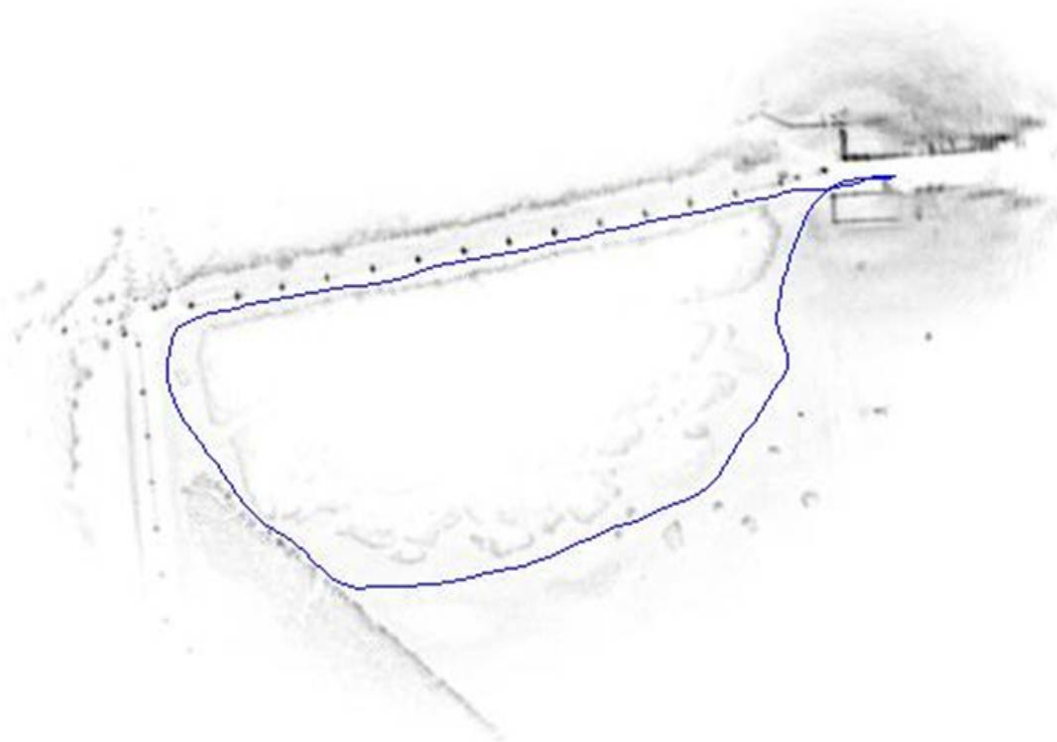


FIGURE 5.8 – Cartographie dense de la zone d’exploration calculée à partir des panoramiques et des positions estimées.

entre le SLAM RS-SLAM-FMT et le SLAM SICK ainsi qu’une représentation, sous la forme d’ellipse, des incertitudes sur les mesures estimées via le RS-SLAM-FMT.

La figure 5.12 montre sous la forme d’un graphique les erreurs, après bouclage, entre la vérité terrain et le SLAM RS-SLAM-FMT. L’erreur maximale est de  $E_{max} = 1,7\text{ m}$ , l’erreur moyenne est de  $E_{moy} = 0,6\text{ m}$  avec un écart type  $\sigma_{erreur} = 0,45\text{ m}$ . Aucun comparatif des erreurs du SLAM SICK n’est réalisé car ce dernier n’a, dans cette expérimentation qu’un rôle de validation visuelle de la consistance des résultats.

Enfin, la figure 5.13 représente une cartographie dense de l’environnement réalisée à partir des panoramiques acquis, des positions estimées par le RS-SLAM-FMT et de la méthode des grilles d’occupation. La zone ne possède que peu d’informations, essentiellement des buissons.

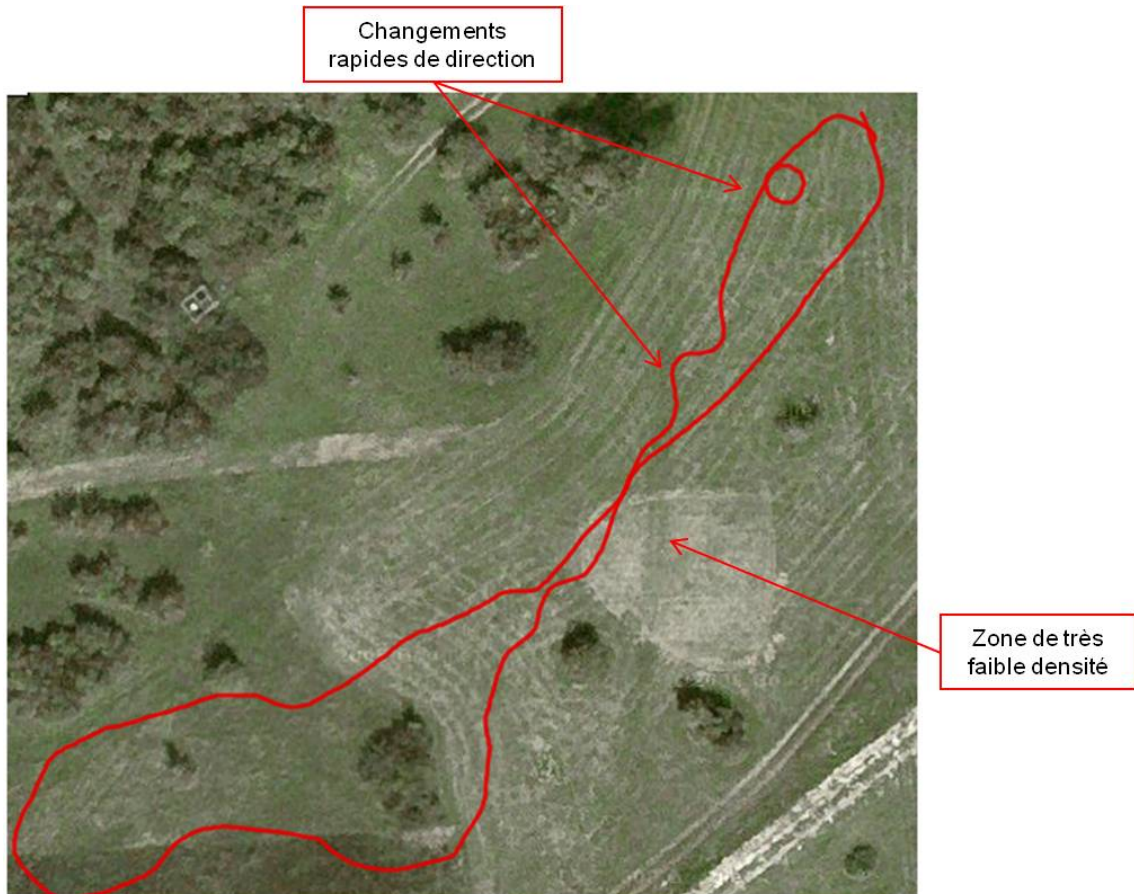


FIGURE 5.9 – Vue aérienne. Le parcours du véhicule est indiqué en rouge.

En conclusion, cet essai montre que l'algorithme RS-SLAM-FMT fournit des résultats en concordance avec d'autres algorithmes de SLAM et d'autres capteurs extéroceptifs (ici un LIDAR). Les changements rapides d'orientation dans une zone à faible densité d'informations ne provoquent pas une mise en échec de l'algorithme. Comme précédemment la vitesse est de l'ordre de 5 km/h et il n'y a aucun objet mobile présent dans la scène.

### 3.4 Conclusion

Les essais dans un environnement naturel avaient pour objectif de valider le fonctionnement du RS-SLAM-FMT dans des conditions de fonctionnement allant de favorables à très défavorables.

Les résultats montrent que le RS-SLAM-FMT, associé au radar K2Pi, fonctionne lorsque l'on explore, à faible vitesse, une zone naturelle. Les problèmes de pente, de changement d'orientation, d'exploration d'environnements clairsemés ou surchargés en points d'intérêts, n'ont pas amené l'algorithme à la faute.

On peut néanmoins regretter que des limites de l'approche n'ont pu être établies. En effet, quelle densité minimum de points d'intérêts faut-il pour mettre en échec le processus ? Quelle pente ou variation brusque d'orientation serait susceptible de faire décrocher l'algorithme ? Ces questions devront trouver une réponse lors d'essais futurs.

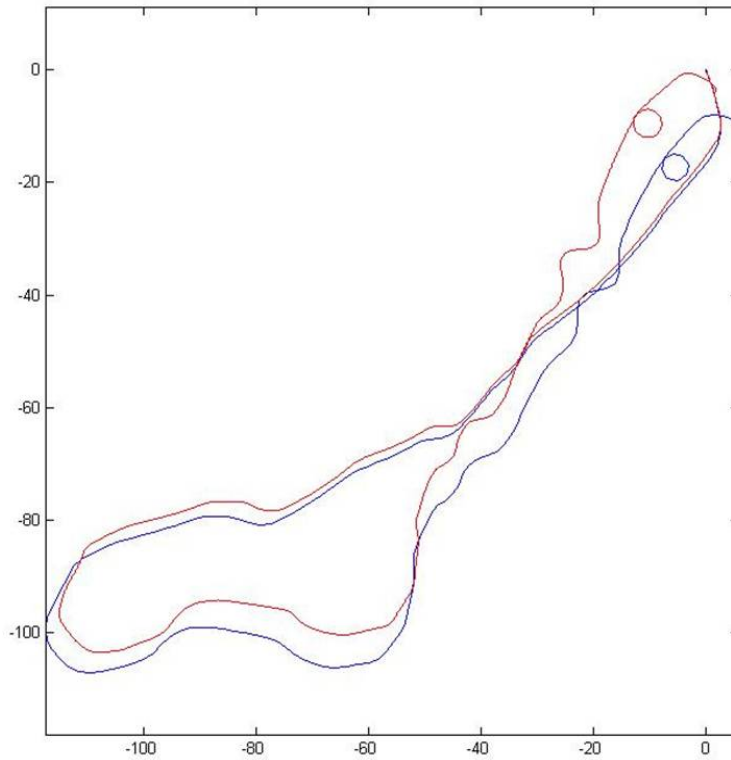
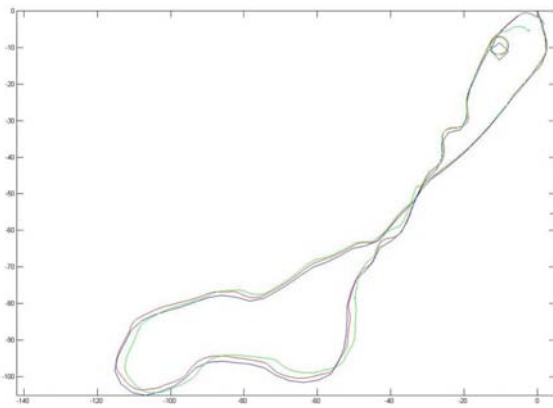
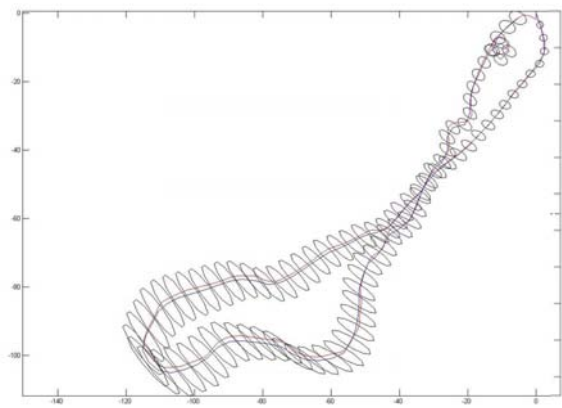


FIGURE 5.10 – En rouge la vérité terrain (GPS RTK), en bleu l'estimation du parcours par une simple prédiction fournie par les proprios.



En rouge la vérité terrain, en bleu le RS-SLAM-FMT, en vert le SLAM SICK.



En rouge la vérité terrain, en noir les ellipses représentant les incertitudes sur les mesures en fin de parcours.

FIGURE 5.11 – Tracés comparatifs du parcours « mouvements brusques »

L'essai suivant va répondre à une autre question. Peut-on utiliser, dans un contexte environnemental favorable (péri-urbain, sans objets mobiles dans notre cas), le système avec un véhicule circulant à une vitesse moyenne de 35 km/h ?

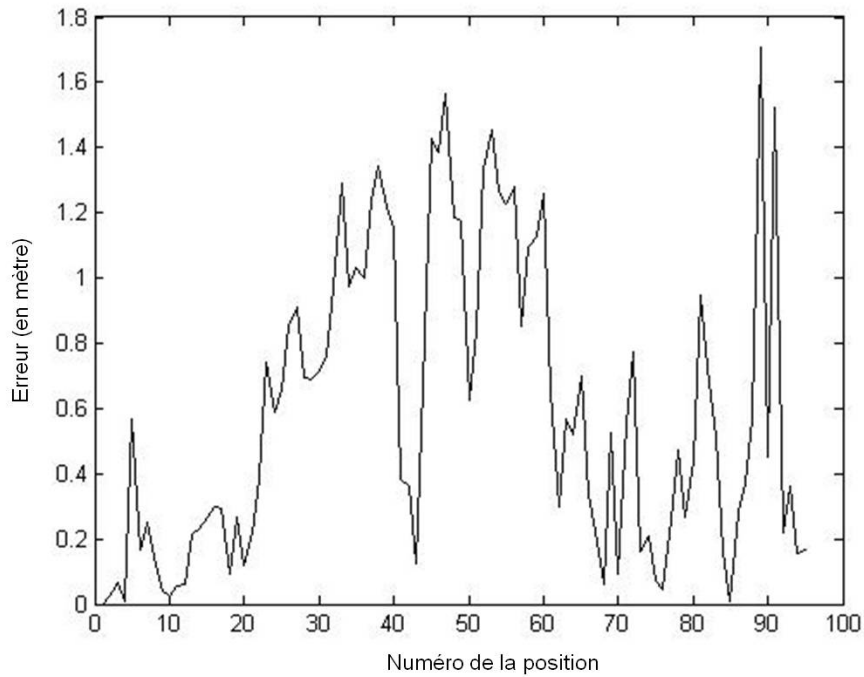


FIGURE 5.12 – Erreurs entre la vérité terrain et le parcours estimé par le SLAM RS-SLAM-FMT.



FIGURE 5.13 – Cartographie dense de la zone d’exploration calculée à partir des panoramiques et des positions estimées.

## 4 Expérimentations en environnement péri-urbain sans objets mobiles

### 4.1 Introduction

Ces essais furent réalisés avec un véhicule du LASMEA nommé « velac » (cf. figure 5.14). On fixe sur son toit, centré sur le train avant, le radar K2Pi. Ce véhicule explore ensuite des zones péri-urbaines à faible densité d'objets mobiles. L'expérimentation présentée a été réalisée sur le parking du zénith de Clermont-Ferrand (cf. figure 5.15) qui est une zone riche en points d'intérêts de formes géométriques simples, dont le sol est plan et qui au moment des essais ne présentait aucun objet mobile.

L'objectif de cet essai est de tester les aptitudes du couple « radar K2Pi, algorithme RS-SLAM-FMT » à réaliser une localisation et cartographie simultanées de bonne qualité dans un environnement favorable, à une vitesse moyenne de 35 km/h et avec de nombreux changements d'orientation. Comme précédemment le SLAM est effectué en post-traitement.



FIGURE 5.14 – Velac : véhicule de test du LASMEA.

### 4.2 Description

La figure 5.15 montre une vue aérienne de la zone, avec le parcours effectué par le véhicule. Les points verts indiquent la présence de lampadaires, à haut pouvoir de réflexion.

Les caractéristiques de l'essai sont résumées dans le tableau 5.3. Elles indiquent notamment que le système réalise la sommation de 3 panoramiques pour l'élimination du speckle et que la vitesse moyenne est de 35 km/h.



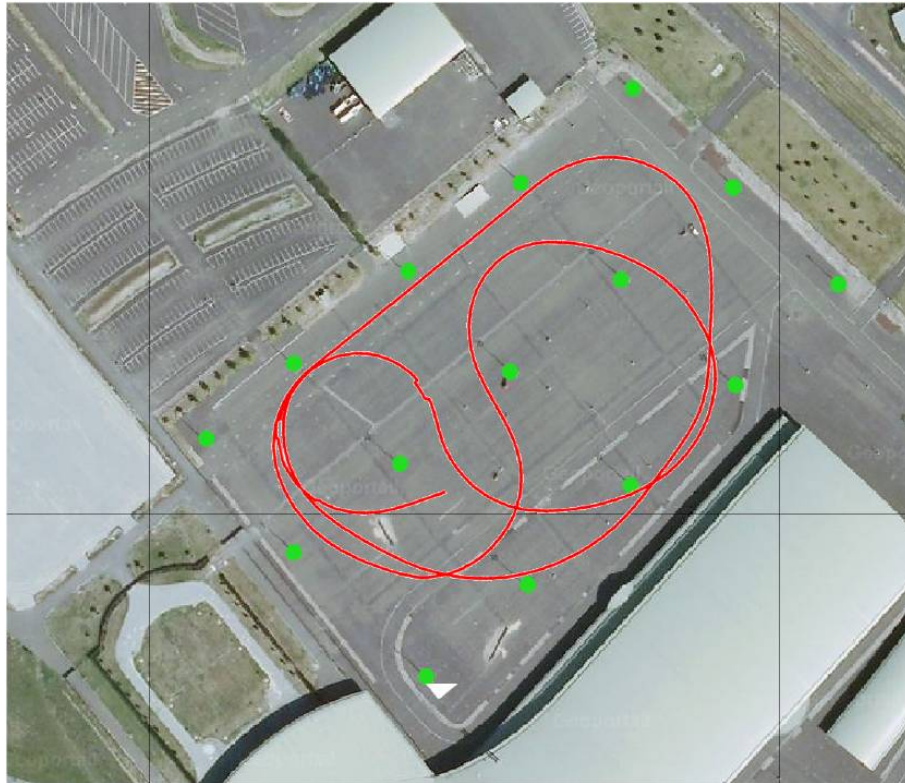


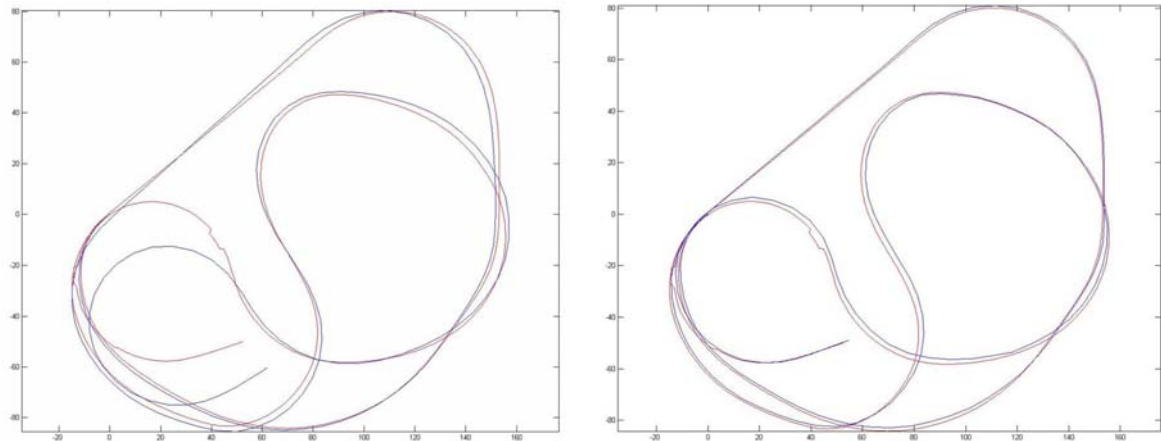
FIGURE 5.15 – Vue aérienne du parcours. En rouge, la vérité terrain (GPS RTK).

TABLE 5.3 – Caractéristiques de l’essai « zenith ».

caractéristiques radar	
Nombre de panoramiques	167
Temps de cycle	1 s
Nombre de panoramiques sommés pour réduction du speckle	3
Déconvolution	OUI
caractéristiques du parcours	
Distance	1230 m
Bouclage	OUI
Pente	NON
Zones denses	OUI
Zones de faible densité	NON
caractéristiques RS-SLAM-FMT	
Distance minimum entre panoramiques	0.5 m
Vitesse moyenne	30 km/h
Gestion des boucles	OUI

La figure 5.16 présente, sur sa partie gauche, l’estimation du parcours par l’utilisation seule des données proprioceptives (odomètre et gyromètre). Sur sa partie droite, elle présente le même parcours estimé par le RS-SLAM-FMT associé au radar K2Pi. On note une grosse erreur du gyromètre en fin de parcours. Cette dernière provoque une divergence très importante de la prédiction alors que le SLAM, avec ses observations, compense ce problème.





En rouge la vérité terrain, en bleu l'estimation des positions par une prédiction via les capteurs proprioceptifs.

En rouge la vérité terrain, en bleu le parcours estimé par le RS-SLAM-FMT.

FIGURE 5.16 – Estimation du parcours.

La figure 5.17 montre, sous la forme d'un graphique, les erreurs après bouclage entre la vérité terrain et le SLAM RS-SLAM-FMT. L'erreur maximale est de  $E_{max} = 3,4 \text{ m}$ , l'erreur moyenne est de  $E_{moy} = 0,94 \text{ m}$  avec un écart type  $\sigma_{erreur} = 0,6 \text{ m}$ .

La figure 5.18 représente une cartographie dense de l'environnement réalisée à partir des panoramiques acquis, des positions estimées par le RS-SLAM-FMT et de la méthode des grilles d'occupation.

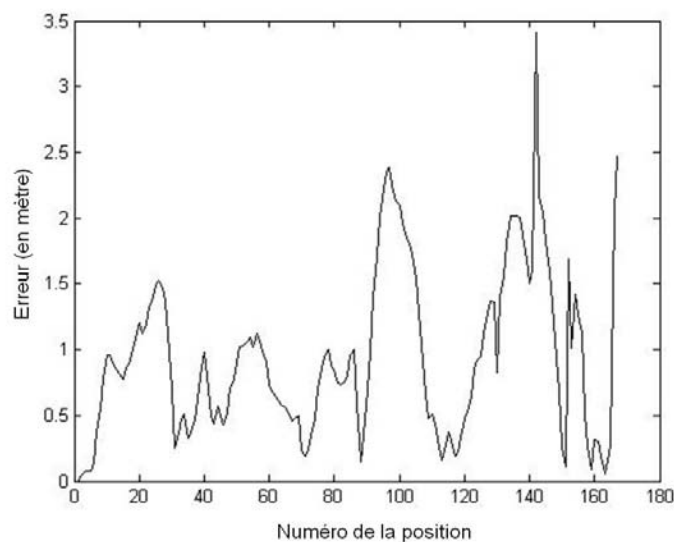


FIGURE 5.17 – Erreurs entre la vérité terrain et le parcours estimé par le SLAM RS-SLAM-FMT.

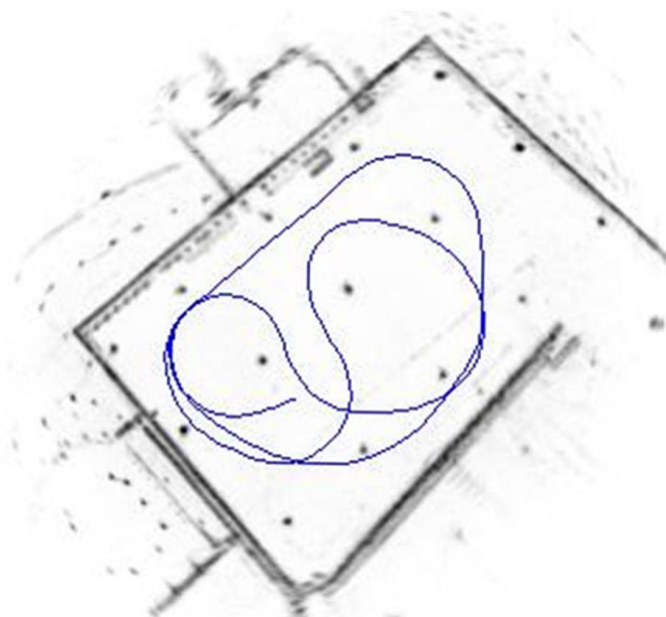


FIGURE 5.18 – Cartographie dense de la zone d’exploration calculée à partir des panoramiques et des positions estimées.

### 4.3 Conclusion

En conclusion, cet essai montre que l’algorithme RS-SLAM-FMT, associé au radar K2Pi, fonctionne lorsque l’on explore, à une vitesse moyenne de 35 km/h, une zone péri-urbaine sans objets mobiles. Les changements d’orientation, ainsi qu’une erreur importante du gyromètre n’ont pas amené l’algorithme à la faute.

La figure 5.19 montre une superposition d’une image aérienne et de la carte dense reconstruite. Aucune étude qualitative n’est réalisée car la problématique principale de ce travail est centrée sur la détermination du parcours et non sur la précision de la carte dense reconstruite.

Il reste maintenant à vérifier que l’algorithme, associé au radar K2Pi, peut fonctionner dans des conditions « extrêmes », c’est à dire dans un environnement péri-urbain à forte proportion d’objets mobiles et avec une vitesse de déplacement conforme à la législation (50 km/h).



FIGURE 5.19 – Superposition d’une vue aérienne (IGN) et de la cartographie dense construite à partir des panoramiques radar et des positions estimées.

## 5 Expérimentations en environnement péri-urbain avec objets mobiles

### 5.1 Introduction

L’étude décrite dans ce document n’a pas pour objectif de détecter des objets mobiles. Le véhicule peut néanmoins être amené à circuler dans un environnement qui en possède. On doit donc vérifier si ce paramètre modifie ou non les aptitudes de l’algorithme RS-SLAM-FMT à faire de la localisation et cartographie simultanées.

Comme dans la section 4, ces essais ont été réalisés avec un véhicule du LASMEA.

La différence entre les deux essais présentés ci-dessous se situe au niveau des conditions d’acquisitions plus ou moins favorables. Le premier se déroule dans un environnement plan, à une vitesse qui n’excède pas les 35 km/h, et avec une densité faible d’objets mobiles. Au contraire, le second se déroule dans un environnement avec dénivelé, à une vitesse qui peut atteindre les 55 km/h, et avec une densité d’objets mobiles forte.

### 5.2 Premier essai en présence d’objets mobiles

Cet essai se déroule sur route ouverte, dans une zone urbaine (cf. figure 5.20), à une vitesse de circulation qui ne dépasse pas les 35 km/h.

Les caractéristiques de l’essai sont résumées dans le tableau 5.4. La figure 5.21 montre quelques clichés de la zone traversée, avec notamment la présence d’objets



FIGURE 5.20 – Vue aérienne du parcours (IGN) de la zone urbaine traversée. En rouge le parcours effectué (données GPS).

mobiles (des véhicules circulant dans la zone).

TABLE 5.4 – Caractéristiques du premier essai en conditions réelles.

caractéristiques radar	
Nombre de panoramiques	128
Temps de cycle	1 s
Nombre de panoramiques sommés pour réduction du speckle	3
Déconvolution	OUI
caractéristiques du parcours	
Distance	1064 m
Bouclage	NON
Pente	NON
Zones denses	quelques objets mobiles
Zones de faible densité	NON
caractéristiques RS-SLAM-FMT	
Distance minimum entre panoramiques	0.5 m
Vitesse moyenne	30 km/h
Gestion des boucles	NON

La figure 5.22 présente quatre figures :

- les variations de vitesse du véhicule tout au long du parcours ;
- l'estimation du parcours par le RS-SLAM-FMT associé au radar K2Pi, avec en rouge la vérité terrain fournie par un GPS-RTK. On note que l'estimation est légèrement biaisée. Ce problème est provoqué par un léger biais continu du gyromètre qui influe sur la construction des panoramiques et en définitive



Début du parcours avec quelques véhicules mobiles présents.



Milieu de parcours, on croise un véhicule.



Milieu de parcours, on croise trois véhicules.



Fin de parcours, on passe à côté d'un grand mur susceptible de saturer le radar.

FIGURE 5.21 – Photographies des zones traversées.

sur l'estimation des positions du véhicule. Dans un parcours en boucle ou non rectiligne cet effet est compensé ;

- le graphique des erreurs qui confirme la remarque précédente, avec une erreur qui ne fait que croître ;
- une représentation des ellipses d'incertitude. Le système ne boucle pas, l'incertitude ne fait donc que croître.

La figure 5.23 représente une cartographie dense de l'environnement réalisée à partir des panoramiques acquis, des positions estimées par le RS-SLAM-FMT et de la méthode des grilles d'occupation. Elle est de moins bonne qualité que dans les essais précédents. Cela est dû au fait, qu'à une vitesse « élevée » et sans rebouclage, une zone de la carte n'est vue que peu de fois. La grille d'occupation ne possède alors que de peu d'informations pour définir la probabilité de présence des objets.

En conclusion, cet essai montre que l'algorithme RS-SLAM-FMT, associé au radar K2Pi, fonctionne lorsque l'on explore, à une vitesse moyenne de 30 km/h, une zone péri-urbaine avec objets mobiles. Les changements d'orientation, n'ont pas amené l'algorithme à la faute. On voit néanmoins apparaître de façon plus importante que dans les essais à basse vitesse des problèmes de biais au niveau de



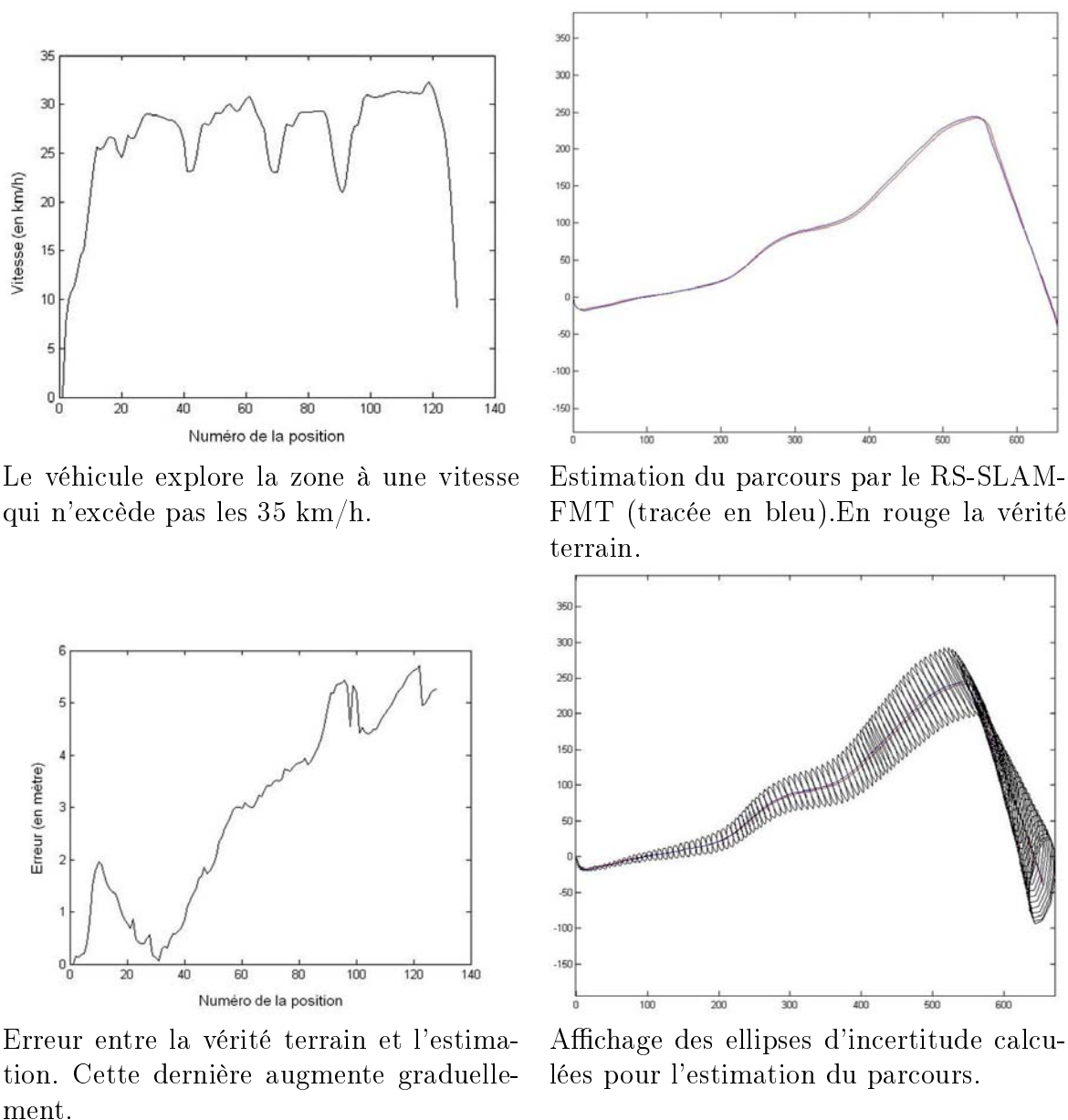


FIGURE 5.22 – Résultats du SLAM effectué sur le premier essai en conditions réelles.

l'orientation.

Reste à vérifier dans le deuxième essai que l'algorithme peut, dans une zone très dense en objets mobiles, continuer à fournir de bons résultats.

### 5.3 Deuxième essai avec objets mobiles

Cet essai se déroule sur route ouverte, dans une zone urbaine (cf. figure 5.24), à une vitesse de circulation qui ne dépasse pas les 55 km/h.

Les caractéristiques de l'essai sont résumées dans le tableau 5.5. La figure 5.25 montre quelques clichés de la zone traversée, avec notamment la présence de nombreux objets mobiles (des véhicules circulant dans la zone) dont un tramway.

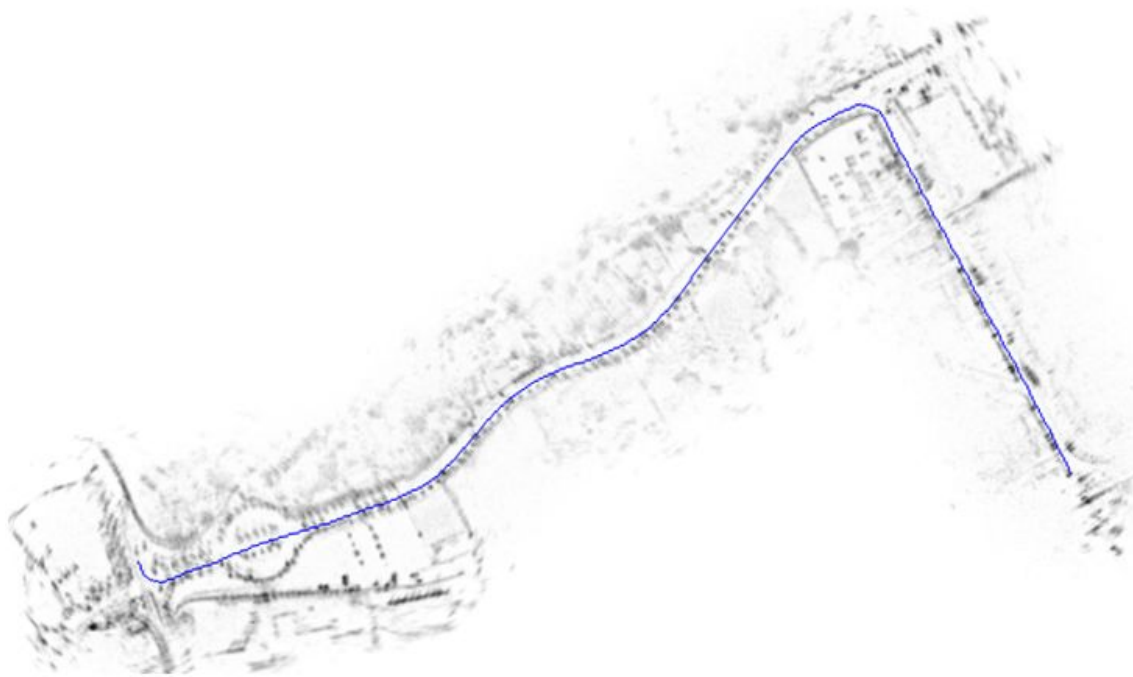


FIGURE 5.23 – Cartographie dense de la zone d’exploration calculée à partir des panoramiques et des positions estimées.

TABLE 5.5 – Caractéristiques du deuxième essai en conditions réelles.

caractéristiques radar	
Nombre de panoramiques	145
Temps de cycle	1 s
Nombre de panoramiques sommés pour réduction du speckle	3
Déconvolution	OUI
caractéristiques du parcours	
Distance	1250 m
Bouclage	NON
Pente	OUI
Zones denses	de nombreux objets mobiles
Zones de faible densité	NON
caractéristiques RS-SLAM-FMT	
Distance minimum entre panoramiques	0.5 m
Vitesse moyenne	45 km/h
Gestion des boucles	NON

La figure 5.26 présente tout d’abord le parcours estimé par une prédiction via les capteurs proprioceptifs. On note que le gyromètre introduit de nombreuses erreurs. Puis, le parcours estimé par le RS-SLAM-FMT associé au radar K2Pi, en rouge la vérité terrain fournie par un GPS-RTK. L’estimation est meilleure qu’avec une simple prédiction, elle reste néanmoins très imprécise (c’est ce que confirme le graphique des erreurs qui montre que le procédé diverge). Le RS-SLAM-FMT reste dépendant

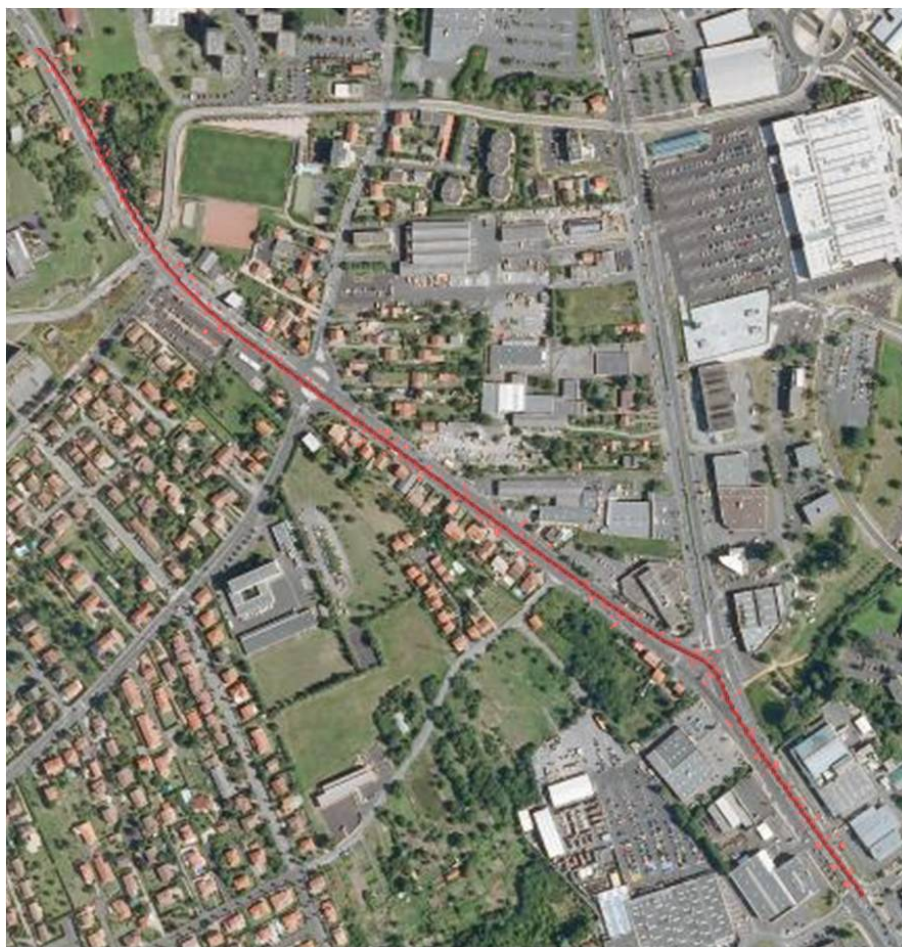


FIGURE 5.24 – Vue aérienne (IGN) de la zone urbaine traversée. En rouge le parcours effectué (données GPS).

du gyromètre, une forte erreur de ce dernier associée au fait qu'à cette vitesse on dispose d'un faible nombre d'observations pour chaque estimation impliquent que l'on ne peut entièrement compenser la divergence. Le dernier graphique donne une représentation des ellipses d'incertitude sur les positions estimées.

La figure 5.27 représente une cartographie dense de l'environnement réalisée à partir des panoramiques acquis, des positions estimées par le RS-SLAM-FMT et de la méthode des grilles d'occupation. Comme pour le premier essai, elle est de mauvaise qualité. Aux conditions précédentes se rajoute le fait que l'estimation est fautive. La carte n'est donc pas révélatrice de l'allure réelle de l'environnement.





Début du parcours avec de nombreux objets mobiles présents sur le boulevard.



Milieu de parcours, la voie est en pente, il y a toujours de nombreux objets mobiles.



Milieu de parcours. L'environnement urbain est en pente. Le véhicule est à l'arrêt aux feux. Le tramway traverse la voie de circulation. De nombreux véhicules sont à l'arrêt pour un court instant.

FIGURE 5.25 – Photographies des zones traversées.

En conclusion, cet essai montre que l'association d'une vitesse rapide, de nombreux objets mobiles et d'un environnement en pente provoque la divergence de l'algorithme RS-SLAM-FMT. Il faut toutefois rappeler que les erreurs importantes du gyromètre ont créé des panoramiques faussés et à terme des appariements faux. Le radar K2Pi et le RS-SLAM-FMT sont donc dépendants de la qualité du gyromètre. Les précédents essais avaient montré que les défauts pouvaient être corrigés si l'on possède un grand nombre de panoramiques à comparer. En revanche, dans le cas de déplacements rapides, on possède moins de panoramiques pour une zone donnée, les comparaisons sont moins nombreuses et les erreurs provoquées par le gyromètre sont donc proportionnellement plus difficilement compensables.

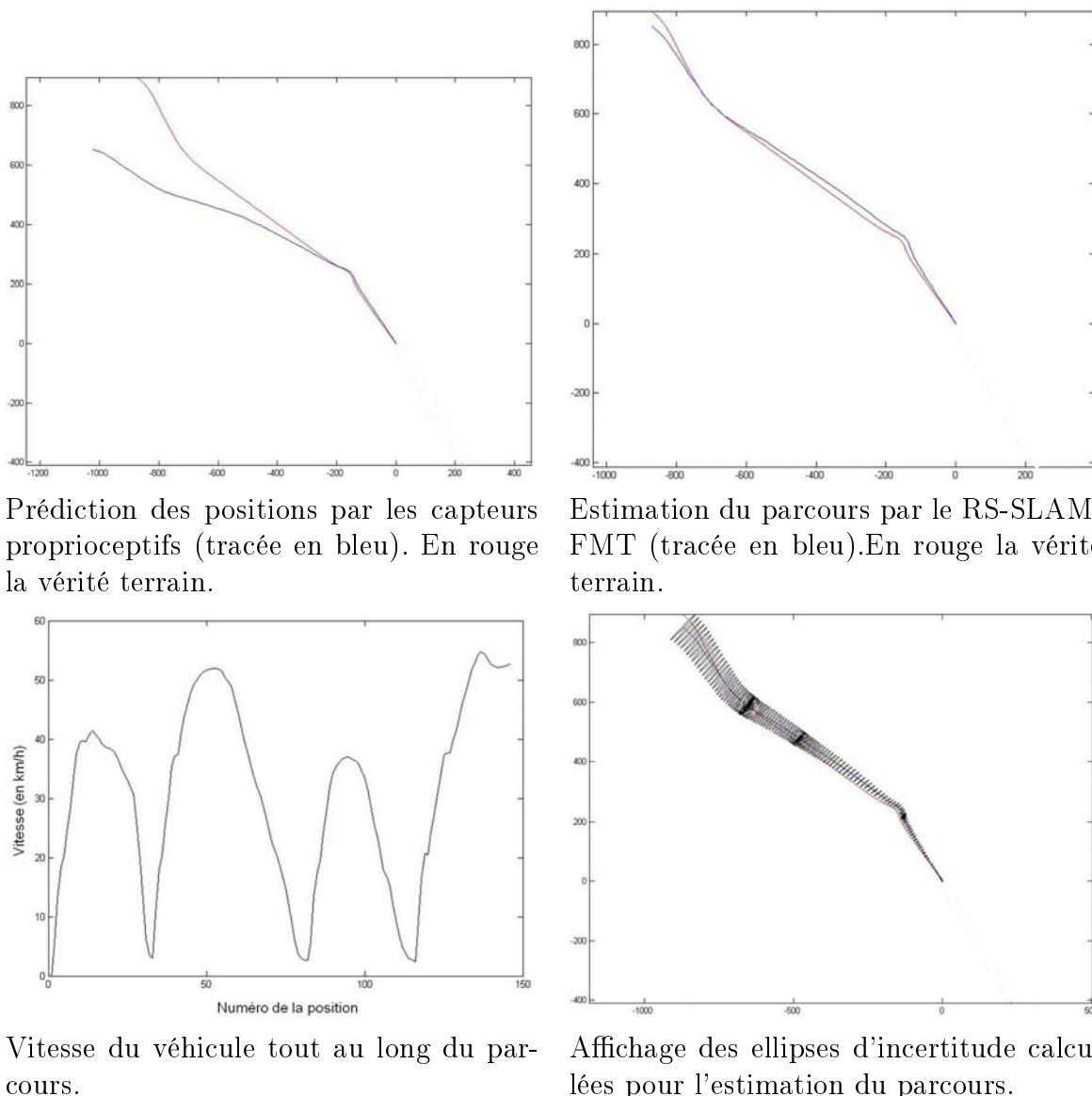


FIGURE 5.26 – Résultats du SLAM effectué sur le deuxième essai en conditions réelles.

## 6 Analyse des essais

Les essais présentés ont eu pour objectif, d'une part de valider l'approche vis à vis d'autres algorithmes de SLAM, et d'autre part de mettre en œuvre l'algorithme dans des conditions et des environnements divers.

Dans le cadre de l'ANR IMPALA, la validation du bon fonctionnement de la méthode en milieu naturel plus ou moins dense a été une réussite. Le robot d'exploration a traversé, à faible vitesse :

- des zones naturelles ambiguës ;
- des zones clairsemées ;
- des zones urbaines à fort pouvoir de saturation.

Ces mouvements ont été variés, avec ou sans changement brusque d'orientation. Tous ces cas, réalisés sans la présence d'objets mobiles n'ont pas pu mettre en échec

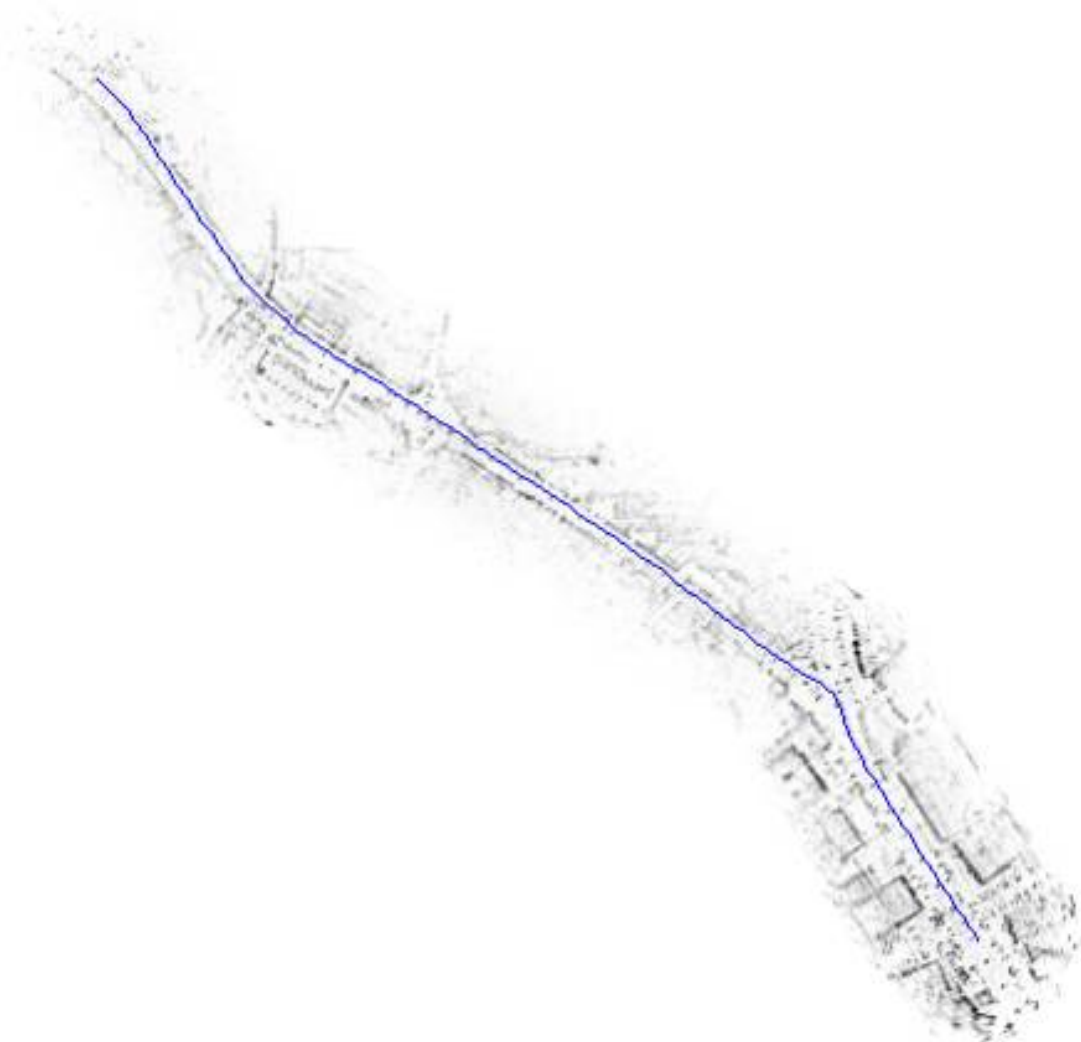


FIGURE 5.27 – Cartographie dense de la zone d’exploration calculée à partir des panoramiques et des positions estimées.

l’algorithme d’estimation.

Une analyse plus poussée serait intéressante afin de déterminer les limites du procédé dans les cas suivants :

- manque de points d’intérêts ;
- degré maximum d’inclinaison du parcours ;
- problèmes de saturation liés à la nature des objets réfléchis ;
- traversée de zones ambiguës telles que les couloirs.

Dans les tests plus généraux, c’est-à-dire dans un cadre péri-urbain avec une vitesse d’exploration « normale », l’algorithme a su montrer ses aptitudes. Néanmoins, une zone dense, très fournie en objets mobiles et traversée à haute vitesse (55 km/h) a su le mettre en échec.

Il reste donc encore du travail à réaliser pour déterminer la dépendance de l’al-

gorithme aux erreurs de construction des panoramiques. Ces erreurs dépendent de plusieurs facteurs :

- qualité des capteurs proprioceptifs ;
- méthode utilisée pour corriger les problèmes de distorsion liés à l'effet Doppler ;
- technique mise en œuvre pour déconvoluer les signaux radar.

Dans ce cadre, une première étude a été réalisée sur un simulateur. Son objectif a été de montrer l'influence des capteurs proprioceptifs sur la construction des panoramiques et sur la comparaison de ces derniers.

## 6.1 Influence des capteurs proprioceptifs sur les observations

Comme l'explique le chapitre 2, la construction d'un panoramique nécessite la connaissance du trajet effectué par le radar lors de l'acquisition des 360 faisceaux. Pour cela, on dispose d'un codeur qui donne la position angulaire de chaque faisceau dans le repère radar. Reste à positionner ces faisceaux dans le repère global. Pour cela, on utilise une technique qui calcule les mouvements du radar à partir de la connaissance des valeurs odométriques et gyrométriques lors de l'acquisition. Si ces valeurs sont bruitées, le résultat est forcément influencé, reste à savoir de quelle manière.

La figure 5.28 représente le parcours radar et les bâtiments qui constituent la base de la simulation radar. Étant en mode simulé, ce dernier fournit une vérité terrain, et des valeurs de l'odomètre et du gyromètre parfaites. Lors de ce parcours, 445 acquisitions sont réalisées (signaux radar, odomètre, gyromètre, GPS).

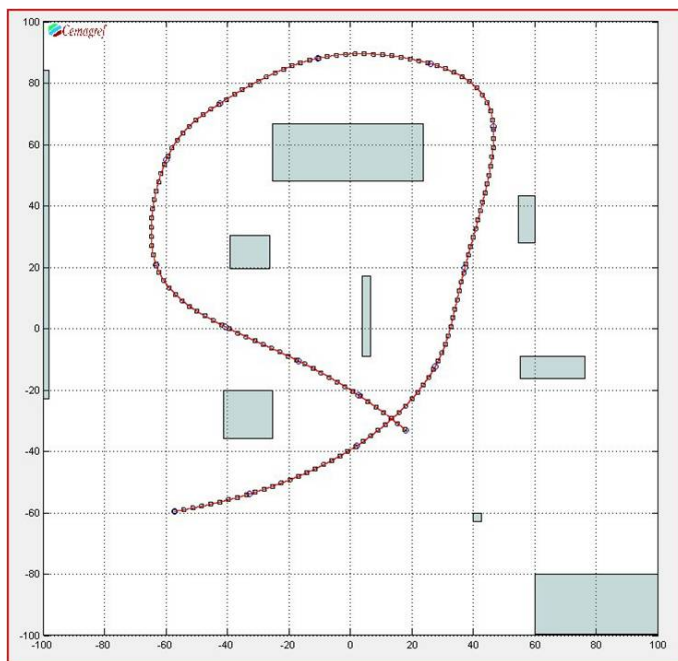


FIGURE 5.28 – Parcours simulé avec un environnement type péri-urbain.

La méthodologie utilisée pour déterminer l'influence des capteurs proprioceptifs sur les panoramiques a été la suivante :

- bruitage des données proprioceptives ;

- construction des panoramiques ;
- recherche par la méthode FMT de la transformation rigide entre panoramiques voisins ;
- calcul des erreurs par rapport à la vérité terrain ;
- tracé des erreurs en translation et en rotation.

Dans ce cadre, plusieurs expérimentations ont été effectuées :

- acquisition et construction des panoramiques sans bruitage des capteurs proprioceptifs, puis calcul des transformations rigides. La figure 5.29 fournit les erreurs en translation et en rotation ;
- acquisition et construction des panoramiques avec bruitage du gyromètre par un bruit gaussien d'écart type  $1^\circ$ . La figure 5.30 fournit les erreurs en translation et en rotation ;
- acquisition et construction des panoramiques avec bruitage de l'odomètre par un bruit blanc d'écart type  $0,3\text{ m}$ . La figure 5.31 fournit les erreurs en translation et rotation ;
- acquisition et construction des panoramiques avec bruitage, du gyromètre par un bruit blanc d'écart type  $1^\circ$ , et de l'odomètre par un bruit blanc d'écart type  $0,3\text{ m}$ . La figure 5.32 fournit les erreurs en translation et rotation ;

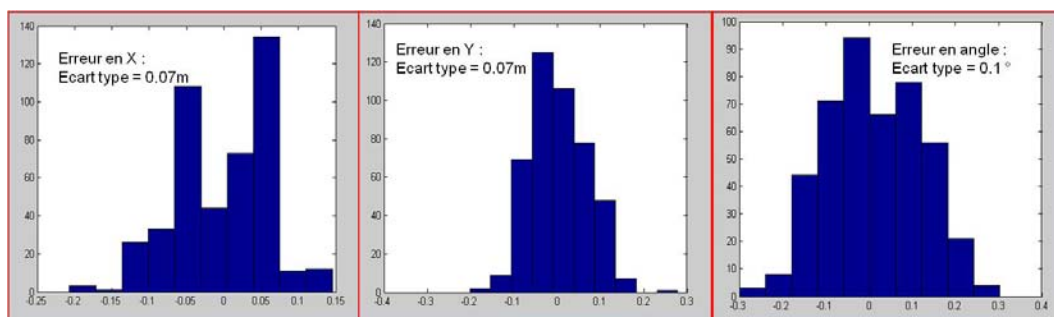


FIGURE 5.29 – Distribution des erreurs en  $(\Delta x, \Delta y, \Delta \theta)$  des transformations rigides entre panoramiques voisins. Les capteurs proprioceptifs ne sont pas bruités, les erreurs sont issues des calculs numériques et de la discrétisation choisie.

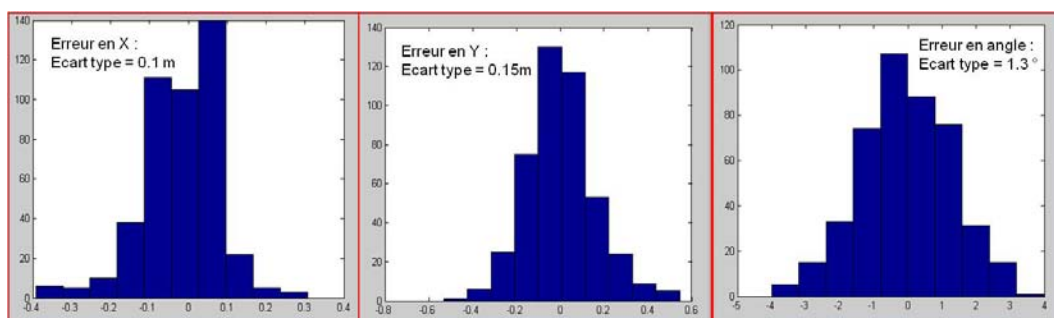


FIGURE 5.30 – Distribution des erreurs en  $(\Delta x, \Delta y, \Delta \theta)$  des transformations rigides entre panoramiques voisins. Le gyromètre est bruité avec un bruit blanc d'écart type  $1^\circ$  qui influence directement la précision d'orientation de la transformation rigide. L'erreur  $\Delta \theta$  est fonction du bruit du gyromètre.

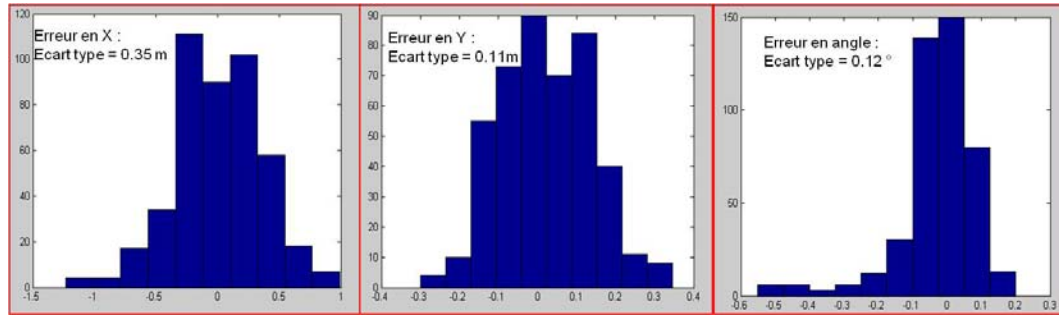


FIGURE 5.31 – Distribution des erreurs en  $(\Delta x, \Delta y, \Delta \theta)$  des transformations rigides entre panoramiques voisins. L'odomètre est bruité avec un bruit blanc d'écart type  $0.3 \text{ m}$  qui influence directement la précision de la translation calculée dans la transformation rigide. L'erreur  $\Delta x$  est fonction du bruit de l'odomètre.

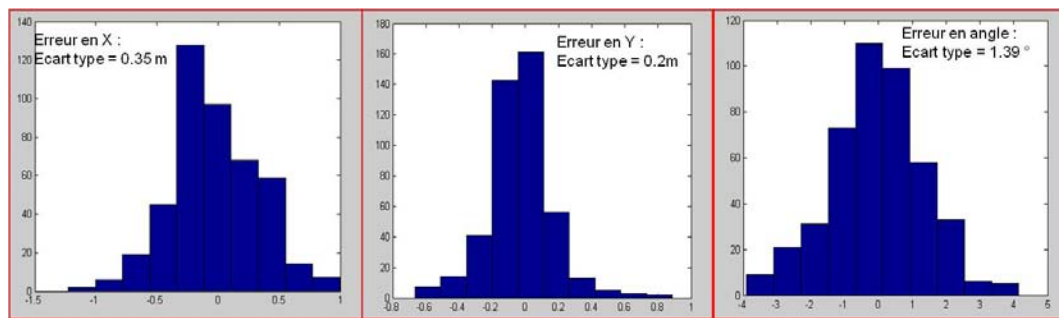


FIGURE 5.32 – Distribution des erreurs en  $(\Delta x, \Delta y, \Delta \theta)$  des transformations rigides entre panoramiques voisins. L'odomètre est bruité, le gyromètre est bruité. La translation et l'orientation sont directement dépendantes de ces erreurs.

En conclusion, si on réalise une seule mise en correspondance par mise à jour de l'algorithme RS-SLAM-FMT, sa qualité sera directement liée à la précision des capteurs proprioceptifs et n'apportera donc aucune plus value par rapport à une simple prédiction. Par contre, si pour chaque mise à jour, plusieurs mises en correspondance sont possibles, l'influence du bruit des capteurs sur la mesure en sera d'autant réduite.

## 7 Comparaison RS-SLAM-FMT et ROLAM

### 7.1 Introduction

La conclusion sur les essais en environnement extérieur a montré que les panoramiques radar sont dépendants de la qualité des capteurs proprioceptifs. De ce fait, lorsque le RS-SLAM-FMT ne dispose que d'un nombre faible de comparaisons entre les panoramiques, la nouvelle position estimée est fortement dépendante des données proprioceptives.

Pour résoudre cela, en parallèle des recherches sur le RS-SLAM-FMT, d'autres travaux se sont attachés à trouver une méthode pour s'affranchir de l'odomètre et du gyromètre. Cette méthode permet d'extraire les vitesses linéaires et angulaires du

véhicule à partir des données radar. Dénommée ROLAM<sup>2</sup>, elle permet alors d'estimer l'évolution du véhicule et d'en déduire son trajet et la carte de l'environnement traversé.

Cette section va donc comparer un parcours de quelques centaines de mètres estimé par le RS-SLAM-FMT et par le ROLAM.

## 7.2 Principe du ROLAM

L'utilisation d'un capteur télémétrique tournant en robotique mobile à haute vitesse implique l'apparition de distorsions sur les données collectées. Un tel effet est, dans la majorité des études, ignoré ou considéré comme un bruit et, de ce fait, corrigé en utilisant des capteurs proprioceptifs (cas du RS-SLAM-FMT) ou des systèmes de localisation.

Le ROLAM [Vivet et al., 2012] analyse la distorsion pour en extraire l'information du déplacement du véhicule, sans aucun autre apport que celui fourni par le capteur extéroceptif. L'idée consiste à effectuer de la vélocimétrie en analysant la distorsion dans les mesures (cf. figure 5.37). Cette analyse aboutit à des estimations des vitesses angulaires et linéaires du robot mobile. Celles-ci sont ensuite utilisées dans le but de construire, sans aucun autre capteur, la trajectoire du véhicule et la carte radar de l'environnement extérieur exploré.

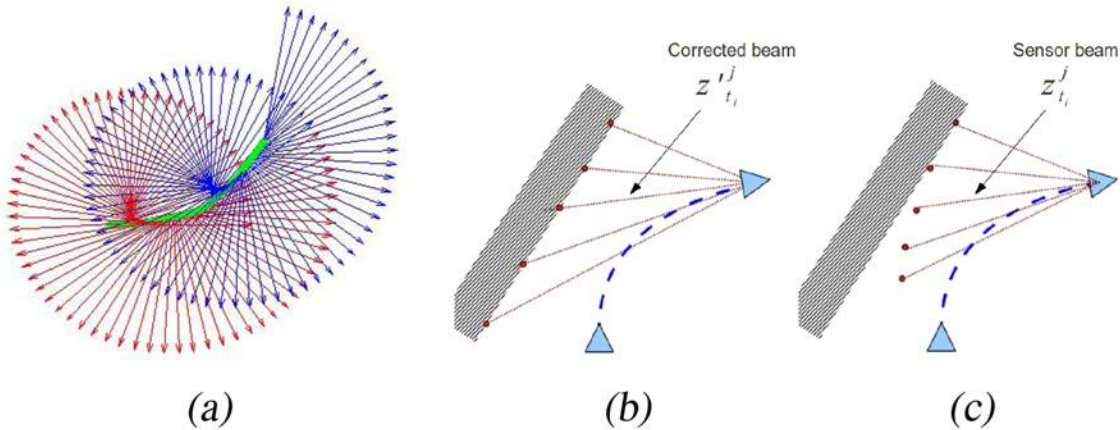


FIGURE 5.33 – (a) Phénomène de distorsion : lorsque le véhicule se déplace selon la trajectoire verte, le capteur effectue une acquisition. Chaque faisceau du capteur est représenté en rouge et en bleu pour, respectivement, la première et la seconde acquisition. Le premier et le dernier faisceau de chaque acquisition ne mesurent pas la même chose. Chaque balayage est distordu par le mouvement. (b) représente l'acquisition souhaitée depuis la seconde position (ou dans le cas d'une approche « stop & scan ») avec les faisceaux corrigés  $z_{t_i}^j$  du scan  $j$  pris au temps  $t_i$ . (c) montre l'acquisition obtenue sans prendre en compte le mouvement c'est-à-dire en considérant les faisceaux  $z_{t_i}^j$  non corrigés [Vivet et al., 2012].

En fait, le déplacement du radar pendant une révolution entière du capteur peut être comparé au mouvement de la valve d'une roue de bicyclette dans un

2. Radar-Only Localization and Mapping



simple déplacement en ligne droite (cf. figure 5.34). En utilisant cela, on peut définir un modèle d'évolution du centre du radar et l'utiliser pour déterminer les données proprioceptives en fonction de la détection d'amers au sein de faisceaux successifs (cf. figure 5.35).

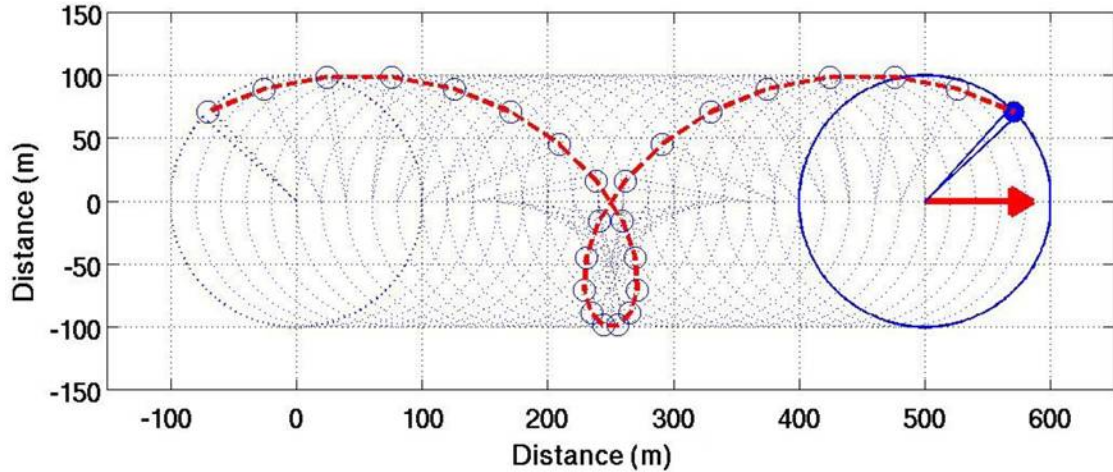


FIGURE 5.34 – Représentation de la position d'un faisceau radar tout au long d'un déplacement en ligne droite [Vivet et al., 2012].

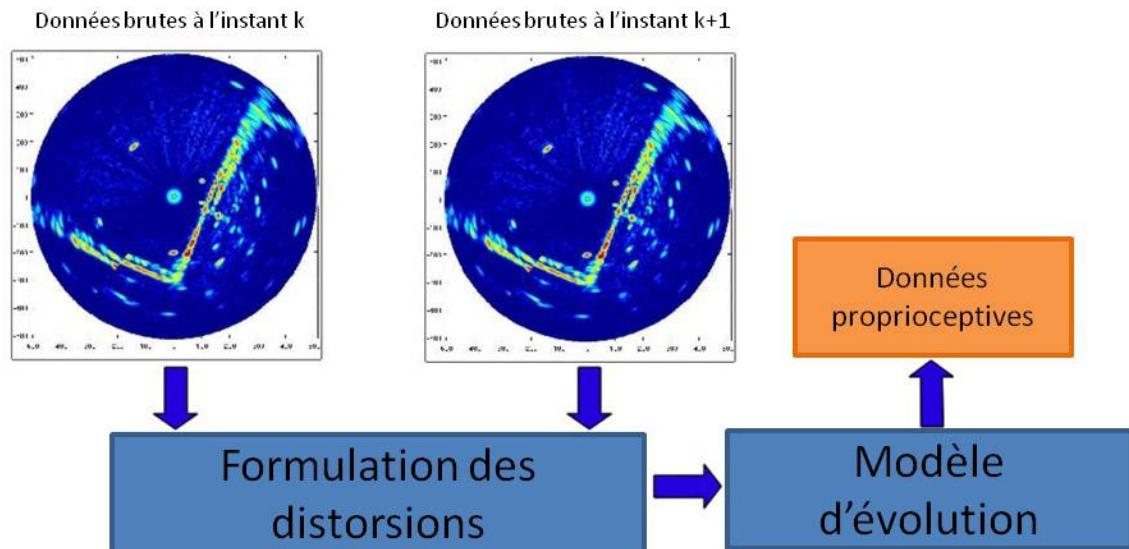


FIGURE 5.35 – Description du procédé d'estimation des données proprioceptives à partir des faisceaux radar [Vivet et al., 2012].

### 7.3 Essai comparatif

Cette section donne les résultats des expérimentations pour les deux approches. Le radar, les capteurs proprioceptifs et un GPS RTK (il fournit des données considérées comme la vérité terrain) sont montés sur Velac (cf. figure 5.14), qui est un des véhicules d'expérimentation du LASMEA. La zone explorée se situe au zénith



d'Auvergne (cf. figure 5.38). L'environnement est constitué de formes géométriques variées, il n'y a pas d'objets mobiles et la vitesse moyenne du véhicule est de  $10\text{ m/s}$ .

Les trajectoires estimées par les deux algorithmes sont données sur la figure 5.36. Le tracé des erreurs de distance entre les positions GPS et les positions estimées est donné figure 5.37.

Le ROLAM présente une erreur initiale importante due au fait que lors de l'accélération du véhicule, le modèle d'évolution qui suppose une vitesse constante rentre en défaut. On note des erreurs assez similaires entre les deux méthodes. Il faut noter qu'à cette vitesse, le RS-SLAM-FMT ne peut réaliser au maximum que 2 comparaisons de panoramiques par mise à jour. Cela signifie qu'il est très dépendant des capteurs proprioceptifs.

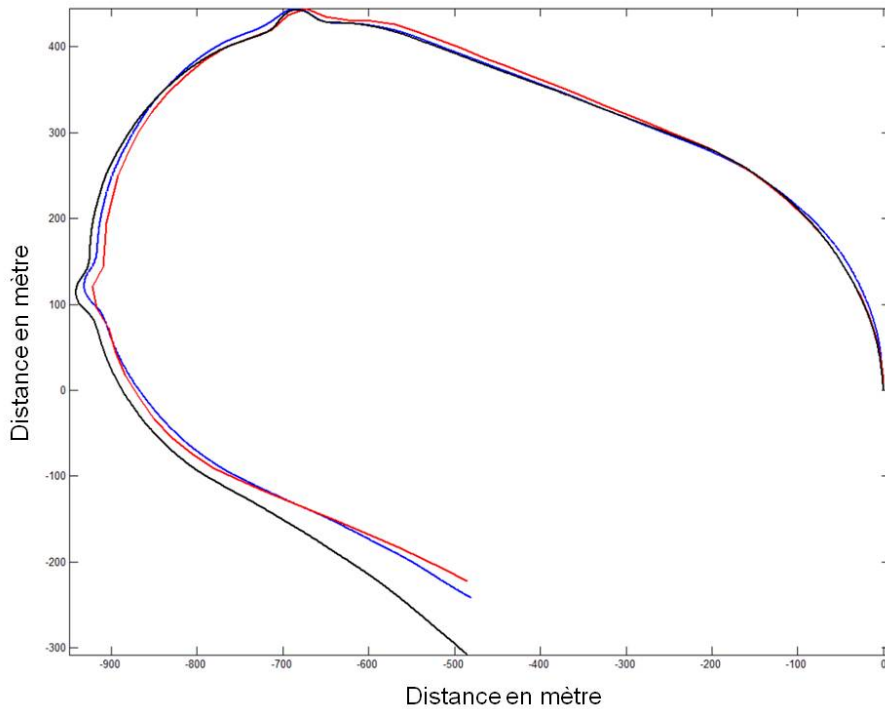


FIGURE 5.36 – Les trajectoires obtenues par les deux méthodes. En noir, la vérité terrain (GPS RTK), en rouge la trajectoire obtenue par le RS-SLAMT-FMT, et en bleu celle obtenue par le ROLAM [Vivet et al., 2012].

Pour quantifier de façon plus précise la qualité des algorithmes, on recherche les erreurs de déplacement et d'orientation pour chaque pose. On obtient, comme erreur moyenne de déplacement, une valeur  $E_{moy} = -0,09\text{ m}$  avec un écart type de  $\sigma_{erreur} = 0,38\text{ m}$  pour le RS-SLAM-FMT, et une erreur moyenne  $E_{moy} = -0,20\text{ m}$  avec un écart type de  $\sigma_{erreur} = 0,70\text{ m}$  pour le ROLAM. Au niveau des erreurs d'orientation on obtient une erreur moyenne  $E_{moy} = -0,03^\circ$  avec un écart type de  $\sigma_{erreur} = 0,1^\circ$  pour le RS-SLAM-FMT et une erreur moyenne  $E_{moy} = 0^\circ$  avec un écart type de  $\sigma_{erreur} = 0,15^\circ$  pour le ROLAM.

L'analyse de ces valeurs montre que les deux méthodes fournissent des estimations correctes de la trajectoire du véhicule. On peut alors construire les cartes à partir des trajets estimés (la figure 5.39 pour le RS-SLAM-FMT et la figure 5.40 pour le ROLAM).

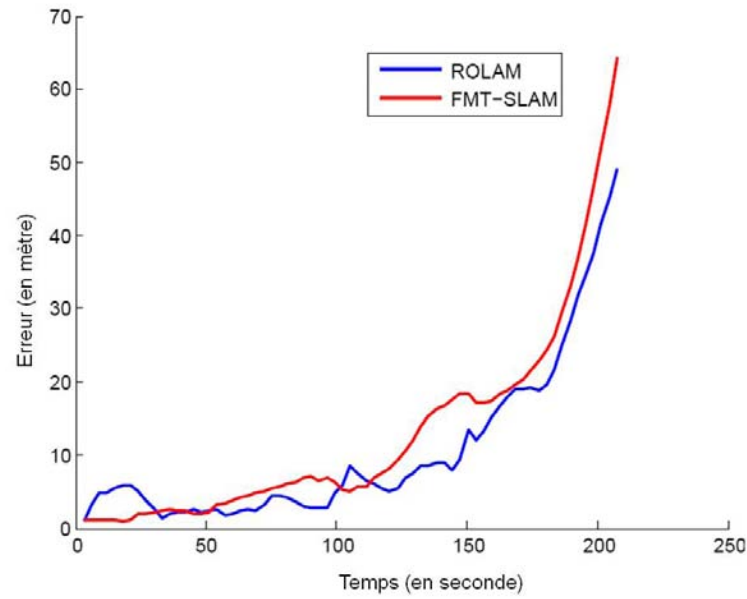


FIGURE 5.37 – Erreurs de distance entre les positions estimées et les positions vraies (GPS RTK) [Vivet et al., 2012].

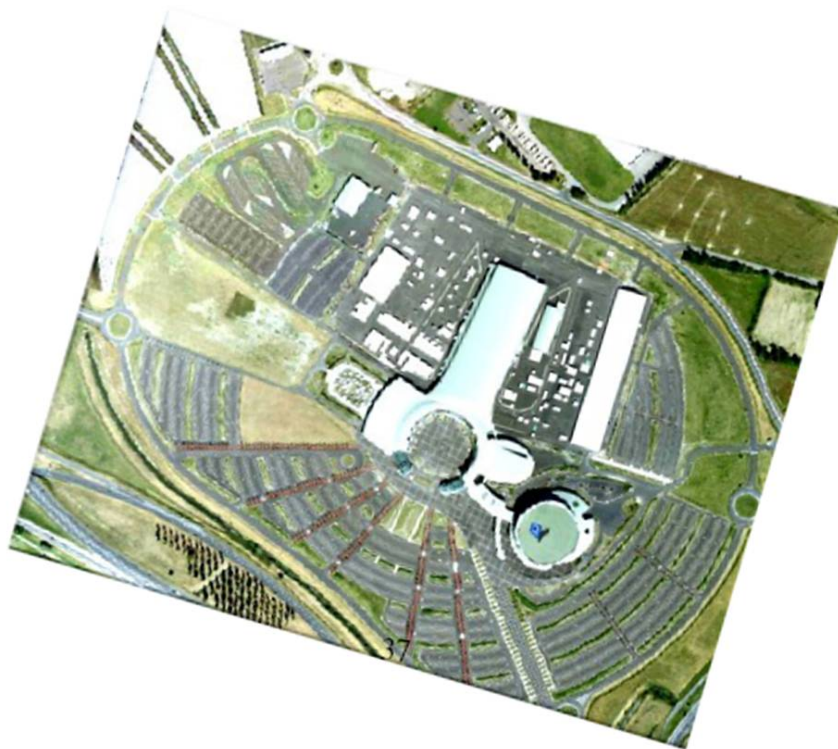


FIGURE 5.38 – Vue aérienne de la zone d'expérimentation.

## 7.4 Conclusion

L'utilisation d'un capteur tournant en robotique mobile à haute vitesse implique l'apparition de distorsions sur les données collectées. Le RS-SLAM-FMT résout ce

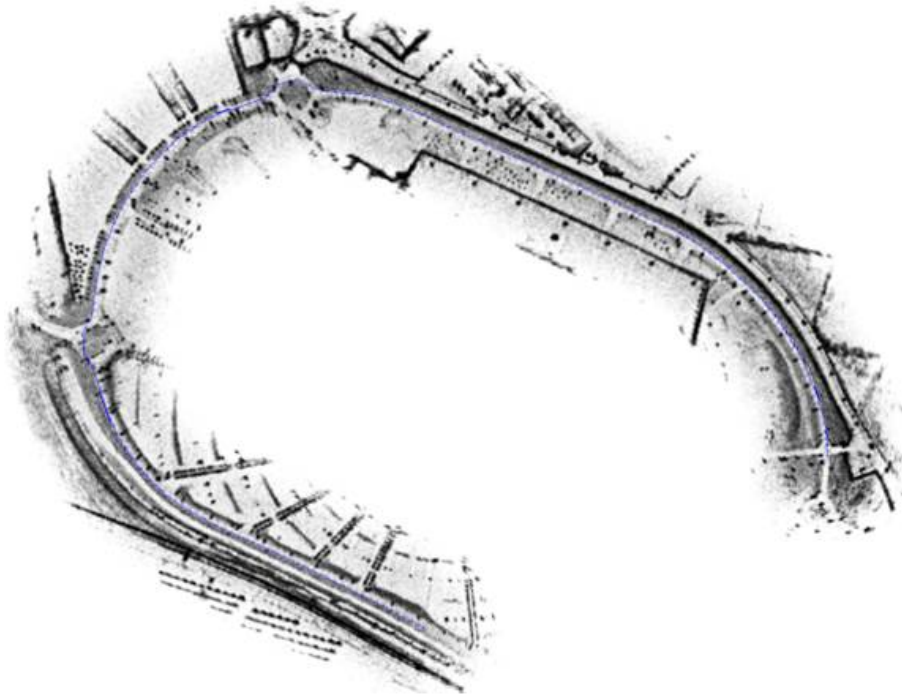


FIGURE 5.39 – Cartographie obtenue par le RS-SLAM-FMT.

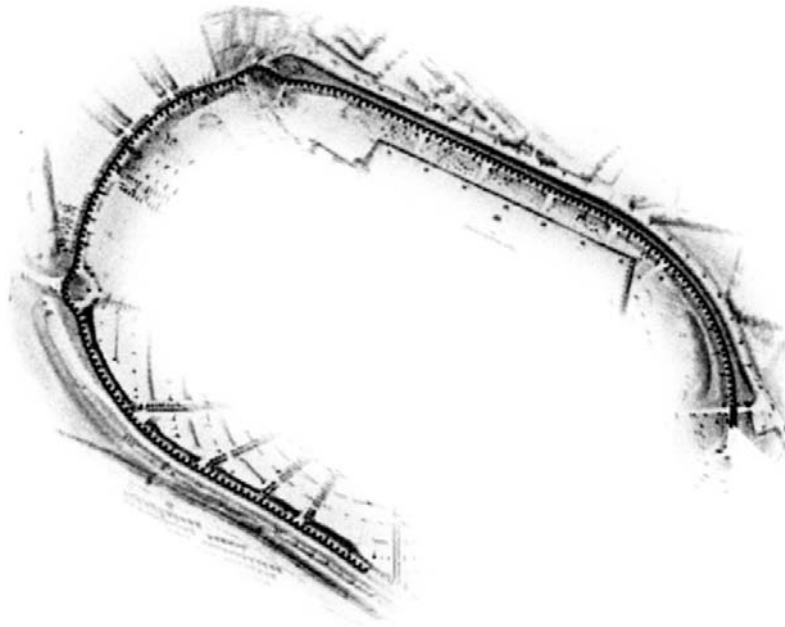


FIGURE 5.40 – Cartographie obtenue par le ROLAM [Vivet et al., 2012].

problème en utilisant des informations issues de capteurs proprioceptifs. Cette dépendance, importante à haute vitesse lorsque l'on ne peut utiliser qu'une ou deux observations, est gênante et difficile à quantifier.

A cette vitesse, le ROLAM permet d'estimer les vitesses linéaires et angulaires simplement par l'extraction d'amers dans les faisceaux radar.

Lors d'expérimentations sur un trajet de quelques centaines de mètres, le ROLAM et le RS-SLAM-FMT donnent des résultats très similaires. Il faut néanmoins rappeler que le ROLAM ne fait pas du SLAM, il ne réalise pas la sauvegarde ni d'amers ni de scans en mémoire, il se contente d'estimer entre faisceaux proches les vitesses linéaires et angulaires du véhicule. Il les utilise ensuite pour effectuer de la prédiction.

De ce fait, un bouclage n'apportera pas d'amélioration sensible du procédé qui continuera inexorablement à diverger. Cela étant dit, on peut tout de même noter qu'il permet d'obtenir des informations proprioceptives de bonnes qualités.

Il apparaît donc intéressant, dans le cadre de travaux futurs, de considérer ce procédé comme une technique qui peut permettre d'éliminer les capteurs proprioceptifs de la chaîne de construction des panoramiques, et par relation d'éliminer l'influence des données proprioceptives sur le *scan matching*.

## 8 Implémentation temps réel

### 8.1 Introduction

Après l'analyse en robustesse du RS-SLAM-FMT dans des expérimentations variées, nous allons aborder le dernier point de ces travaux.

Notre projet a pour objectif de tester l'utilisation du radar K2Pi dans des applications de SLAM « statique » et d'en réaliser l'intégration temps réel. Le portage devra à terme être implanté sur le R-Trooper, véhicule d'exploration de la société THALES, et sera validé lors d'essais sur des sites d'évaluation.

Dans ce contexte, le choix d'utiliser un algorithme SLAM utilisant un filtre de Kalman a été fait. De cette façon, à chaque nouvelle observation, le RS-SLAM-FMT peut fournir une estimation de la nouvelle position et une information sur son incertitude. Il reste à déterminer si un *scan matching* par *Transformée de Fourier-Mellin*, associé à un EKF-SLAM peut espérer respecter un fonctionnement temps réel. Dans le cas précis de l'utilisation du radar K2Pi, cela correspond à des temps de calculs inférieurs à la seconde (en effet le radar fournit des panoramiques toutes les secondes).

### 8.2 Implémentation temps réel

Le R-Trooper et son architecture multiprocesseur sont les éléments qui devront accueillir l'application SLAM temps réel. Dans ce contexte on peut décrire l'interface fonctionnelle mise en œuvre (cf. figure 5.41). Pour plus de renseignements on peut se reporter à l'annexe C.

Les différentes étapes du portage de l'algorithme RS-SLAM-FMT sont :

- codage en C++ des fonctionnalités et test sur une plateforme C++ multi-processeurs ;

- intégration des fonctionnalités dans une DLL et test sur une plateforme Visual C++ ;
- mise en œuvre de la DLL sur RTMaps, qui est la plateforme de développement installé sur le R-Trooper, avec des données en mode « rejoue » ;
- test en live sur le véhicule R-Trooper.

Le schéma de principe pour la mise en œuvre du procédé est donné figure 5.41. Il montre que la programmation est séparée en deux parties distinctes, d’une part la construction des panoramiques et d’autre part le RS-SLAM-FMT. Il est à noter que la partie construction de la cartographie dense n’est pas réalisée dans cette intégration. La figure présente également les entrées/sorties et les informations sur le stockage des données.

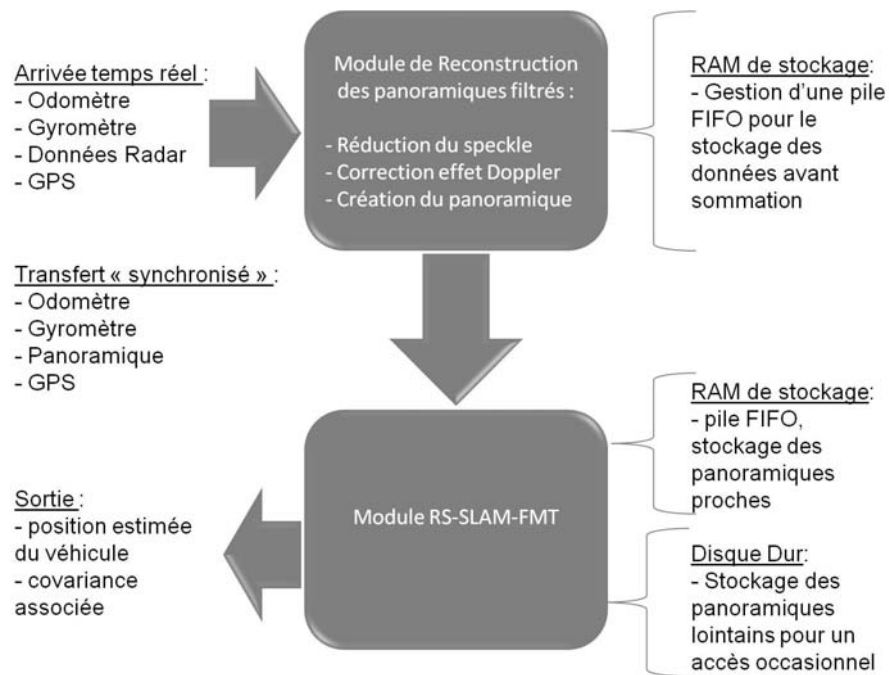


FIGURE 5.41 – Interface fonctionnelle de l’intégration temps réel du RS-SLAM-FMT.

**Module de reconstruction des panoramiques filtrés :** il construit les panoramiques en éliminant le speckle. Son temps de traitement ne doit pas excéder le temps de cycle d’arrivée des faisceaux radar (soit une seconde). Afin d’optimiser son fonctionnement, il dispose d’une pile FIFO de stockage qui permet d’accéder de façon rapide aux faisceaux précédents.

**Module RS-SLAM-FMT :** il calcule la meilleure estimation possible de la position du véhicule. Il utilise pour cela les panoramiques filtrés issus du bloc précédent et les données proprioceptives synchronisées sur les faisceaux radar. Il possède une zone de stockage rapide, constituée d’une pile FIFO, qui permet d’avoir très rapidement accès aux panoramiques les plus proches lorsque l’on réalise le *scan matching*. Dans le cas des boucles, les comparaisons s’effectuent avec des panoramiques très éloignés, ces derniers ont été sauvegardés sur disque dur.

### 8.3 Résultats et conclusion

Des essais avec le véhicule R-Trooper ont été enregistrés. Ils ont ensuite été utilisés en mode rejeu pour fournir à la DLL des conditions de test les plus réalistes possibles.



FIGURE 5.42 – Zone urbaine traversée par le R-Trooper.

L'exemple donné dans cette section représente un test effectué à vitesse réduite ( $3\text{ m/s}$ ) sur une zone industrielle désaffectée de la région parisienne (cf. figure 5.42). On retrouve sur la figure 5.43 le parcours réel (en rouge) et l'estimation de la trajectoire, soit par une simple prédiction (en vert), soit par la DLL du RS-SLAM-FMT (en bleu). Le résultat montre une bonne estimation de la trajectoire en accord avec les résultats obtenus lors de nos tests précédents. On voit également que l'algorithme prend en compte des boucles qui, on le sait, sont très coûteuses en temps de calcul. Les temps de calculs obtenus sont décrits dans le tableau 5.6.

Ces résultats représentent un RS-SLAM-FMT calculé à l'aide de la DLL implémentée sur un ordinateur Dual-Core 2 Ghz. Ils sont identiques à ceux obtenus avec une implémentation `Matlab`, mais bien entendu, ils sont obtenus avec des temps de calculs optimisés.

On voit que la partie reconstruction nécessite  $0.7s$  pour calculer un panoramique filtré. Cela respecte le fait que ce processus doit avoir une durée inférieure au temps de cycle radar (le radar fournit les 360 faisceaux en  $1\text{ s}$ ). En ce qui concerne le temps d'exécution du processus SLAM, ce dernier varie entre  $0.5\text{ s}$  et  $2.7\text{ s}$ . Cette variation est causée par le fait que le SLAM cherche à comparer chaque nouvelle observation avec des observations anciennes. En moyenne, dans l'exemple donné,



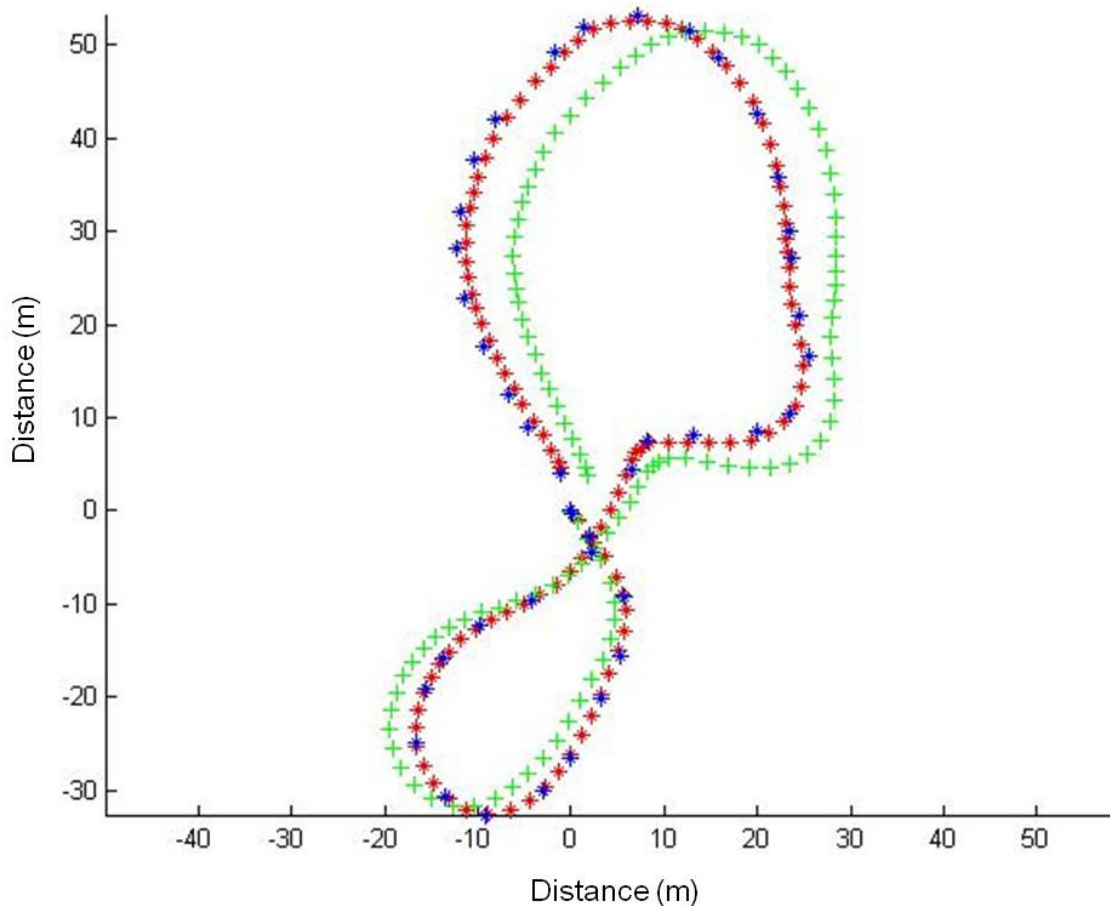


FIGURE 5.43 – RS-SLAM-FMT (en bleu) calculé sur un parcours en boucle (en rouge la vérité terrain). On retrouve en vert le parcours calculé par une simple prédiction.

TABLE 5.6 – Temps de calcul des deux processus principaux.

Nom du <i>thread</i>	Durée min (s)	Durée maxi (s)	Durée moy (s)
Thread construction d'un panoramique filtré	0.7	0.7	0.7
Thread RS-SLAM-FMT	0.5	2.7	0.9

chaque cycle de mise à jour compare la nouvelle observation avec quatre observations anciennes. Néanmoins, en cas de bouclage, le nombre de comparaisons augmente et par conséquent le temps de calcul également.

Par rapport à l'implémentation *Matlab*, en moyenne on diminue par 4 le temps de construction du panoramique filtré, et on diminue par 6 celui du RS-SLAM-FMT.

Le procédé fournit donc des résultats de construction des panoramiques filtrés qui ont un temps de traitement inférieur à 1 s. Par contre le RS-SLAM-FMT a un temps d'exécution qui varie suivant le nombre d'observations comparées. Cela signifie que lors d'une boucle (cas le plus exigeant en temps de calcul, car il nécessite un grand nombre de comparaisons et des accès à des données très anciennes qui sont sauvegardées sur disque dur) on ne respecte plus la règle des 1 s. On va donc perdre quelques panoramiques filtrés qui ne seront pas pris en compte. Il est néanmoins préférable de gérer correctement les boucles quitte à perdre un ou deux panoramiques

et donc une ou deux étapes d'estimation de la position du véhicule.

Il reste à réaliser l'implémentation sur RTMaps afin de vérifier si l'on peut obtenir des résultats équivalents voire meilleurs. Ces tests ont débuté, et les premiers résultats montrent la faisabilité du procédé dans des temps de cycle inférieurs à la seconde.

## 9 Conclusion

Ce chapitre a présenté deux parties distinctes de l'étude. D'une part des tests de robustesse du RS-SLAM-FMT par comparaison avec d'autres algorithmes existants, et via des campagnes d'essais variées. D'autre part le travail effectué en aval pour réaliser une implémentation temps réel de l'algorithme.

La comparaison de l'algorithme estimatif avec une méthode globale du type *Square Root SAM* a montré, lors d'un test de courte distance et sans bouclage, que leurs résultats sont très proches. L'utilisation de l'EKF-SLAM comme algorithme estimatif est possible.

En ce qui concerne les essais de l'algorithme dans des environnements variés. On s'aperçoit que le RS-SLAM-FMT donne de très bons résultats dès lors que l'on respecte une vitesse de croisière faible (inférieure à 30 *km/h*). Des zones naturelles sans amers, des zones péri-urbaines très segmentées, ne le mettent pas en défaut. Par contre, dès l'instant où la vitesse augmente (circulation urbaine à 50 *km/h*), le nombre d'observations à comparer diminue, la qualité des informations proprioceptives baisse, tout cela provoque une augmentation des erreurs de localisation.

Cette influence des capteurs proprioceptifs sur le résultat final est un point faible de notre méthode. Aussi, des tests futurs sont prévus pour remplacer ces capteurs par des techniques d'estimation locale des vitesses linéaire et angulaire tel que le propose le ROLAM [Vivet et al., 2012].

Un dernier phénomène a été testé lors de ces essais, c'est la présence d'objets mobiles dans le champ d'acquisition du radar. Les premiers résultats montrent que la technique utilisée pour supprimer les effets de speckle limite l'influence de ces perturbations (notre SLAM « statique » ne prend pas en compte ces informations, elles sont donc considérées comme du bruit). Seule une densité très forte d'éléments en mouvement met en défaut le RS-SLAM-FMT. Pour améliorer ces tests, il nous reste à quantifier cette influence.

En ce qui concerne le portage des algorithmes, un effort important a été consacré à l'implémentation temps réel de l'algorithme avec portage du code `Matlab` initial en `C++`. Ce travail de plusieurs mois a permis d'obtenir des gains de temps significatifs : le RS-SLAM-FMT se découpe en deux parties, d'une part la création des panoramiques et d'autre part le SLAM lui même. Pour la première partie le gain de temps est d'un rapport 4 (par rapport à son implémentation `Matlab`) et pour la deuxième partie il est d'un rapport 6. Cela permet d'effectuer un SLAM sans bouclage en moins d'une seconde, ce qui permet de respecter le temps d'acquisition cyclique du radar.

Dans le cas précis du bouclage, l'algorithme effectue un nombre de tests plus important, cela augmente les temps de calcul qui dépassent alors la seconde. Le



contrecoup de cette opération est la perte d'un ou deux nouveaux panoramiques avec pour conséquence d'avoir la mise à jour suivante calculée avec un nombre faible de comparaisons. Cela reste acceptable quand on connaît l'importance du bouclage pour améliorer la précision d'une localisation.

En conclusion, la mise en application du radar K2Pi dans un algorithme de localisation et cartographie simultanées fonctionne et devrait, dans les années futures apporter un outil supplémentaire aux véhicules d'explorations. Son implémentation temps réel est d'ores et déjà réaliste et doit permettre une utilisation concrète du procédé.

# Conclusions et perspectives

## Bilan général

L'utilisation d'un radar à onde continue dans un procédé de localisation et de cartographie simultanées a permis de créer un algorithme novateur de SLAM orienté trajectoire nommé RS-SLAM-FMT [Gérossier et al., 2009]. Il est constitué d'une nouvelle méthode de *scan matching* fondée sur la *Transformée de Fourier-Mellin*, d'une technique originale d'estimation de l'incertitude sur la transformée trouvée, d'une approche EKF-SLAM inédite qui s'applique sur des données de type panoramique radar, et d'un procédé de création d'une cartographie directement utilisable par un opérateur.

Il est important de revenir sur chacun de ces éléments ainsi que sur l'implémentation temps réel de l'algorithme.

### *Scan matching par Transformée de Fourier-Mellin*

Le radar K2Pi fournit des panoramiques qui sont une image de son environnement au moment des acquisitions. Dans ce contexte, une technique de mise en correspondance appelée *scan matching* a été utilisée. Dans les applications SLAM, elle est souvent réalisée via un algorithme ICP. Dans le cas du radar, nous avons montré qu'un panoramique est fortement bruité. Il fournit toutefois plus qu'une simple localisation des objets, il donne également des informations sur leurs pouvoirs réfléchissants. L'utilisation de la méthode ICP, qui ne prend en compte que les positions des objets, montre rapidement ses limites. Nous avons donc développé une nouvelle technique fondée sur la *Transformée de Fourier-Mellin* (FMT). Elle permet la mise en correspondance de panoramiques dans leur ensemble et non plus d'objets localisés. Les informations de réflexion et de similitude des formes sont toutes deux prises en compte.

Pour valider cette approche, une étude comparative entre la méthode ICP et la méthode FMT a été détaillée dans ce manuscrit. Elle montre que, pour des temps de calculs comparables, la FMT permet la recherche des transformations rigides  $[R, T]$  sur des données plus ou moins bruitées, alors qu'ICP se retrouve plusieurs fois mis en échec.

Dans l'application SLAM, le *scan matching* donne à l'algorithme d'estimation les transformations rigides et les matrices de covariance associées. Ces dernières sont souvent déterminées, au préalable, de façon expérimentale et ne donnent qu'une valeur moyenne de l'incertitude. Il apparaît intéressant de pouvoir estimer au mieux cette incertitude pour chaque transformation rigide, cela ne peut qu'améliorer en aval les résultats d'estimation des positions. Dans ce cadre, une solution inédite est proposée dans ce mémoire. Elle est associée à la méthode de mise en correspon-

dance et permet, pour chaque transformation rigide calculée, de donner une valeur d'incertitude sur la translation et sur la rotation.

### **Algorithme d'estimation des positions**

L'algorithme d'estimation choisi dans ce travail s'appuie sur le filtre de Kalman étendu. Il est souvent utilisé dans le cadre d'un SLAM à base d'amers, et sa mise en œuvre dans un SLAM orienté trajectoire avec *scan matching* reste moins courante.

Les travaux présentés dans ce document ont mis en application une adaptation de l'EKF-SLAM au contexte du radar K2Pi. Ils montrent sur des tests en conditions réelles la faisabilité du procédé et sa robustesse face à la diversité de l'environnement, aux conditions climatiques, aux mouvements effectués par le véhicule.

### **Cartographie dense de l'environnement**

Par ailleurs, à partir des positions estimées, le procédé calcule une cartographie dense de l'environnement. Cet outil peut donner des informations complémentaires à un opérateur qui commande un véhicule d'exploration.

Si, par exemple, la zone à explorer est inconnue et ne possède pas de carte *a priori*, l'opérateur en charge de la commande du robot navigue en aveugle. Après quelques secondes de SLAM, l'algorithme d'estimation fournit les positions successives du véhicule, il est donc possible de construire une carte de plusieurs dizaines de mètres de rayon qui renseigne l'opérateur sur les obstacles présents dans la zone. L'opérateur pourrait alors définir un cheminement de navigation en prenant en compte les informations sur l'environnement traversé.

La méthode utilisée pour construire cette carte est fondée sur une superposition de l'ensemble des panoramiques dont les positions et orientations sont stockées dans le vecteur d'état.

### **Implémentation temps réel**

Les avancées scientifiques développées dans ces travaux (création, d'un nouvel algorithme de *scan matching*, d'une technique de calcul des incertitudes sur le matching réalisé, d'un EKF-SLAM utilisant pour la première fois des panoramiques radar) ont été directement appliquées dans le projet ANR IMPALA.

La mise en œuvre avec des données réelles K2Pi montre ainsi le bon fonctionnement du RS-SLAM-FMT. Ces résultats novateurs ont ainsi pu être publiés dans plusieurs conférences internationales [Gérossier et al., 2009 ; Checchin et al., 2010 ; Vivet et al., 2009].

L'intégration du RS-SLAM-FMT sur un démonstrateur temps réel de la société THALES (un des trois partenaires du projet IMPALA) a été effectuée. Le radar K2Pi fournit un panoramique toutes les secondes, les tests montrent que l'algorithme peut traiter ce flux en temps réel.

## **Perspectives**

Les travaux de thèse présentés ont permis de mettre en évidence plusieurs difficultés d'une approche SLAM radar dont la mise en œuvre d'un capteur tournant sur un véhicule en mouvement. Ils ont également montrés que la création d'un cadre

« normalisé » serait très intéressant pour comparer des méthodes de SLAM. Enfin, les résultats obtenus montrent qu’une utilisation du RS-SLAM-FMT sur des parcours de grande distance semble possible. Reste à déterminer les modifications à apporter au RS-SLAM-FMT pour rendre cela possible.

Voici en quelques lignes quelques propositions pour répondre à ces questions.

Une des premières perspectives est la réduction des déformations liées au capteur tournant. A l’heure actuelle nous utilisons les informations fournies par les capteurs proprioceptifs. Elles permettent de calculer les mouvements du radar lors de l’acquisition, et ainsi de les corriger. Néanmoins, des bruits importants sur les capteurs influencent fortement les résultats. Il serait donc intéressant d’utiliser une méthode qui puisse s’affranchir de ces capteurs. Les travaux de Vivet et al. [2012] fournissent un première approche dans ce domaine, avec une estimation des vitesses linéaire et angulaire à partir des seules données radar.

Une autre perspective à court terme concerne la création d’outils de quantification des résultats. Il existe plusieurs pistes, comme par exemple celle de Jaulmes et al. [2009] qui proposent un procédé de quantification des résultats fournis pour tous les types d’algorithmes de SLAM. Les critères jugés sont :

- la précision de la carte reconstruite ;
- la robustesse à un odométrie bruitée ;
- la robustesse à du bruit dans les scans ;
- la capacité à travailler dans un environnement clos ;
- la capacité à gérer correctement les boucles ;
- le temps de calcul ;
- les ressources matériels allouées (processeur, mémoire).

Il est à noter que plusieurs points de cette liste ont été testés lors de nos essais. L’intérêt serait de pouvoir quantifier les résultats d’un comparatif entre plusieurs algorithmes de SLAM utilisant le même jeu de données.

Un autre élément qui n’est généralement jamais quantifié est la précision de la carte reconstruite. Elle n’a souvent pour fonction que d’informer l’opérateur via un écran de contrôle. Son évaluation reste donc très souvent visuelle.

Jaulmes et al. [2009] proposent de définir une valeur numérique qui indique la qualité de la carte reconstruite. Elle est calculée par rapport à des points de contrôle préalablement définis et placés sur la carte réelle (ceci est réalisé par un opérateur).

Schwertfeger et al. [2010] proposent de définir la qualité d’une carte par les critères suivants :

- le taux de couverture quantifiable ;
- le niveau de détail de la reconstruction ;
- la précision de la carte (position des objets dans la carte) ;
- la consistance locale : les erreurs de position des différents groupes d’objets entre eux.

Ils proposent pour cela de placer dans l’environnement des amers facilement identifiables par la carte reconstruite. Cette méthode permettrait une évaluation poussée de la carte reconstruite. Par contre, les éléments identifiables seraient également détectés par le SLAM qui en serait alors influencé (on se retrouverait dans le cas d’un capteur de mesure qui influence le résultat du procédé mesuré).

Notre carte dense est construite selon les principes d'une grille d'occupation, on pourrait donc s'inspirer des travaux de Yairi [2004] qui découpe la carte réelle en cellules qui sont comparées avec les cellules de la grille d'occupation.

On voit donc qu'il existe de nombreuses pistes pour quantifier la qualité d'une carte.

Une dernière perspective concerne l'utilisation du RS-SLAM-FMT sur des parcours longues distances (supérieurs à 5 *km*). Dans ce contexte, deux éléments sont à remettre en question : d'une part la technique de gestion de la carte, et d'autre part l'utilisation d'informations globales.

Notre système nécessite de vérifier si la nouvelle position estimée peut être comparée avec les anciennes. Plus le champ d'expérimentation augmente et plus cette vérification va être coûteuse. Dans ce contexte, une évolution du RS-SLAM-FMT vers une technique de découpage en sous-cartes tels que le proposent Estrada et al. [2005] pourrait *a priori* convenir.

Enfin, on voit lors de nos essais, que lors d'un parcours sans boucle, le procédé diverge, l'erreur de positionnement ne fait qu'augmenter. Même si cette augmentation reste faible, elle va provoquer, dans le cas de longues distances sans bouclage, des erreurs importantes. Il apparaît donc intéressant de pouvoir disposer à certains instants d'un positionnement global. La première idée serait de coupler notre procédé à un GPS « bas coût » qui fournirait, à des instants donnés, une position du véhicule. Une autre idée serait de fusionner la carte du RS-SLAM-FMT avec une carte aérienne fournie par un drone qui réaliserait un SLAM aérien de la zone.

# ANNEXES



# Annexe A

## Algorithme général de l'EKF-SLAM

### 1 Introduction

Cette annexe donne une présentation des variables et équations modélisant un EKF-SLAM par carte d'amers.

Cela débute par une présentation du vecteur d'état et de sa matrice de covariance, puis une explication sur la modélisation des transitions d'état. Vient ensuite la fonction d'observation et enfin la fonction de cartographie qui permet d'affiner le vecteur d'état selon les données d'observation.

### 2 Espace d'état

L'espace d'état correspond à la représentation d'un système par un ensemble de paramètres également appelés variables d'état. L'ensemble de ces  $n$  variables est écrit sous la forme d'un vecteur d'état :

$$\mathbf{X} = [x_1 \dots x_i \dots x_n]^T \quad (\text{A.1})$$

Dans une application SLAM avec carte d'amers, ces paramètres  $x_i$  correspondent d'une part à la position en deux dimensions d'un véhicule et d'autre part à des amers constituant une carte de l'environnement dans lequel évolue le véhicule. Le système d'état est défini par le vecteur suivant :

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_V \\ \mathbf{X}_M \end{bmatrix} \quad (\text{A.2})$$

Le vecteur  $\mathbf{X}_V$  décrit la position du véhicule dans un plan référencé par rapport à un repère global  $L$ . Il se compose de trois états, avec par exemple :  $(x_v, y_v)$  la position cartésienne du centre de l'essieu arrière du véhicule et  $\theta_v$ , l'orientation de celui-ci.  $\mathbf{X}_M$  se rapporte au vecteur d'état de la carte. Le nombre d'états de ce vecteur dépend du nombre d'amers détectés dans l'environnement et de la géométrie associée à chaque amer.

Pour employer un filtre EKF, le vecteur  $\mathbf{X}$  est modélisé par une variable aléatoire suivant une loi gaussienne. Le système (A.2) est représenté par deux moments définissant la loi normale ou gaussienne :



– une moyenne :

$$\hat{\mathbf{X}} = \begin{bmatrix} \hat{\mathbf{X}}_V \\ \hat{\mathbf{X}}_M \end{bmatrix} \triangleq \mathbb{E}[\mathbf{X}] \quad (\text{A.3})$$

– une matrice de covariance :

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_V & \mathbf{P}_{VM} \\ \mathbf{P}_{VM}^T & \mathbf{P}_M \end{bmatrix} \triangleq \mathbb{E} \left[ \left( \mathbf{X} - \hat{\mathbf{X}} \right) \left( \mathbf{X} - \hat{\mathbf{X}} \right)^T \right] \quad (\text{A.4})$$

$\mathbf{P}_V$  représente la matrice de covariance associée au vecteur  $\mathbf{X}_V$  et  $\mathbf{P}_M$  la matrice de covariance associée à  $\mathbf{X}_M$ .  $\mathbf{P}_{VM}$  la matrice de covariance entre les éléments de la carte et les états du véhicule. Cette modélisation de la carte et du véhicule est dite *stochastique* et a été introduite initialement dans l'article [Smith et al., 1990]. Dans la suite de cette section sont introduites les différentes équations permettant d'initialiser, de faire évoluer et d'observer ces différentes variables.

### 3 Transition d'état

Les transitions d'état sont modélisées par la fonction non-linéaire  $\mathbf{f}$ . Cette équation d'évolution du système SLAM (le véhicule et la carte) entre deux instants  $k$  et  $k + 1$  s'écrit :

$$\mathbf{X}(k + 1|k) = \mathbf{f}(\mathbf{X}(k|k), \mathbf{U}(k)) + \mathbf{w}(k) \quad (\text{A.5})$$

$\mathbf{U}(k)$  correspond aux commandes appliquées au véhicule.  $\mathbf{w}(k)$  représente le bruit sur le modèle. L'environnement est représenté par des éléments géométriques. Ces différents éléments, appelés amers, sont considérés comme statiques. La carte n'évolue donc pas entre deux instants :

$$\mathbf{X}_M(k + 1|k) = \mathbf{X}_M(k|k) \quad (\text{A.6})$$

Le seul élément mobile correspond au véhicule. Son évolution est régie par l'équation non-linéaire  $\mathbf{f}_V$  :

$$\mathbf{X}_V(k + 1|k) = f_V(\mathbf{X}_V(k|k), \mathbf{U}(k)) + \mathbf{w}_V(k) \quad (\text{A.7})$$

Le terme  $\mathbf{w}_V$  représente un bruit gaussien sur le déplacement du véhicule et  $\mathbf{U}$  représente les données proprioceptives associées à la commande du véhicule. La fonction  $\mathbf{f}_V$  représente l'évolution des états du véhicule. Elle correspond aux équations cinématiques du modèle du véhicule.

La forme matricielle de l'équation de prédiction des états prend en compte les évolutions du véhicule  $\mathbf{f}_V$  et l'évolution de la carte :

$$\mathbf{X}(k + 1|k) = \begin{bmatrix} f_V(\mathbf{X}_V(k|k), \mathbf{U}(k)) + \mathbf{w}_V(k) \\ \mathbf{X}_M(k|k) \end{bmatrix} \quad (\text{A.8})$$

La matrice de covariance associée au vecteur d'état  $\mathbf{X}(k + 1|k)$  s'écrit :

$$\mathbf{P}(k + 1|k) = \mathbf{F}_X \cdot \mathbf{P}(k|k) \cdot (\mathbf{F}_X)^T + \mathbf{F}_U \cdot \mathbf{Q}_U \cdot (\mathbf{F}_U)^T \quad (\text{A.9})$$

où :

–  $\mathbf{Q}_U$  est la matrice de covariance associée aux bruits  $\mathbf{w}$  ;

- $\mathbf{F}_X$  est la matrice jacobienne de la fonction  $\mathbf{f}$  par rapport à l'état  $\mathbf{X}(k|k)$  :

$$\mathbf{F}_X = \begin{bmatrix} \mathbf{F}_{X_V} & \mathbf{0}_{3 \times (n-3)} \\ \mathbf{0}_{(n-3) \times 3} & \mathbf{I}_{(n-3) \times (n-3)} \end{bmatrix} \quad (\text{A.10})$$

- $\mathbf{F}_U$  est la matrice jacobienne de la fonction  $\mathbf{f}$  par rapport aux données de commande  $\mathbf{U}$  :

$$\mathbf{F}_U = \begin{bmatrix} \mathbf{F}_{V_U} \\ \mathbf{0}_{(n-3) \times 2} \end{bmatrix} \quad (\text{A.11})$$

où  $\mathbf{F}_{V_U}$  correspond à la matrice jacobienne de la fonction  $\mathbf{f}_V$  par rapport à la commande  $\mathbf{U}$ .

## 4 Fonction d'observation

Dans un contexte SLAM, les équations d'observation sont utilisées pour projeter la position estimée des amers dans le repère associé à la position du véhicule. Le but est de comparer, dans un repère commun, la position des amers de la carte estimée avec les mesures recueillies. La fonction d'observation  $\mathbf{h}$  s'écrit :

$$\mathbf{Z}(k+1|k) = \mathbf{h}(\mathbf{X}(k+1|k)) + \mathbf{w}_m(t) \quad (\text{A.12})$$

$\mathbf{w}_m$  représente les bruits gaussiens à moyenne nulle sur les mesures extéroceptives et sur la modélisation des observations. La matrice de covariance associée à cette observation est donnée par :

$$\mathbf{P}_Z(k+1|k) = \mathbf{H}_X \cdot \mathbf{P}(k+1|k) \cdot \mathbf{H}_X^T \quad (\text{A.13})$$

$\mathbf{H}_X$  est la matrice jacobienne de la fonction  $\mathbf{h}$  par rapport à l'état  $\mathbf{X}(k+1|k)$ .

Ces équations d'observation correspondent à des fonctions de composition introduites dans l'article [Smith et al., 1990].

## 5 Fonction de cartographie

On rappelle que l'algorithme général de l'EKF-SLAM a été défini avec un SLAM avec carte d'amers.

Une mesure extéroceptive est modélisée par le vecteur  $\mathbf{Z}_{measure}$  et la matrice de covariance associée  $\mathbf{C}_{measure}$ . Ce vecteur  $\mathbf{P}_{measure}$  contient  $m$  positions d'éléments de l'environnement  $\mathbf{X}_{Ai}$  avec pour chacune, une matrice de covariance associée  $\mathbf{C}_{Ai}$  :

$$\mathbf{Z}_{measure} = \begin{bmatrix} \mathbf{X}_{A1} \\ \vdots \\ \mathbf{X}_{Ai} \\ \vdots \\ \mathbf{X}_{Am} \end{bmatrix} \quad (\text{A.14})$$

Ces positions d'amer  $\mathbf{X}_{Ai}$  sont considérées comme indépendantes les unes des autres. La matrice de covariance associée à la mesure s'écrit donc :

$$\mathbf{P}_{mesure} = \begin{bmatrix} \mathbf{C}_{A1} & \dots & 0 \\ & \ddots & \\ \vdots & & \mathbf{C}_{Ai} & \vdots \\ 0 & \dots & & \mathbf{C}_{Am} \end{bmatrix} \quad (\text{A.15})$$

Lors de la phase d'association de données, la carte estimée  $\hat{\mathbf{X}}_M$  est mise en correspondance avec la carte mesurée. Si un ou plusieurs éléments mesurés ne sont pas associés avec des éléments déjà cartographiés, il sont considérés comme des nouveaux amers. Ces éléments doivent être insérés dans la carte estimée  $\hat{\mathbf{X}}_M$ . Il sont notés  $\mathbf{X}_{nouveau}$  avec comme matrice de covariance associée  $\mathbf{C}_{nouveau}$ .

Les mesures de l'environnement  $\mathbf{X}_{Ai}$  sont référencées par rapport à la position du véhicule  $\mathbf{X}_V$ . La fonction non-linéaire  $\mathbf{g}$  translate ces nouvelles positions dans le repère  $L$  de la carte  $\mathbf{X}_M$  :

$$\mathbf{X}_{nouveau}^L = \mathbf{g}(\mathbf{X}(k+1|k), \mathbf{X}_{nouveau}) + \mathbf{w}_m(k) \quad (\text{A.16})$$

$\mathbf{w}_m$  représente les bruits gaussiens à moyenne nulle sur les mesures extéroceptives. Ensuite, le vecteur d'état du nouvel amer est inséré dans le vecteur d'état  $\mathbf{X}$  du système :

$$\mathbf{X}_{aug} = \begin{bmatrix} \mathbf{X} \\ \mathbf{X}_{nouveau} \end{bmatrix} \quad (\text{A.17})$$

Le calcul de la matrice de covariance augmentée  $\mathbf{P}_{aug}$  doit prendre en compte les incertitudes sur la mesure de la position du nouvel amer  $\mathbf{C}_{nouveau}$  et l'incertitude de la position du véhicule dans la carte estimée  $\mathbf{P}_V$ . Cette matrice est obtenue avec les calculs :

$$\mathbf{P}_{aug} = \mathbf{G} \cdot \begin{bmatrix} \mathbf{P} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{nouveau} \end{bmatrix} \cdot \mathbf{G}^T \quad (\text{A.18})$$

$\mathbf{G}$  représente la jacobienne de la fonction  $\mathbf{g}$  par rapport aux vecteurs  $\mathbf{X}$  et  $\mathbf{X}_{nouveau}$  :

$$\mathbf{G} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{G}_{X_V} & \mathbf{G}_{X_{nouveau}} \end{bmatrix} \quad (\text{A.19})$$

$\mathbf{G}_{X_V}$  correspond à la matrice jacobienne de la fonction  $\mathbf{g}$  par rapport à  $\mathbf{X}_V$  et  $\mathbf{G}_{X_{nouveau}}$  à la matrice jacobienne de la fonction  $\mathbf{g}$  par rapport au vecteur  $\mathbf{X}_{nouveau}$ .

# Annexe B

## Essais en conditions réelles

### 1 Introduction

Dans le cadre du projet ANR IMPALA (cf. chapitre 9), une évaluation des méthodes et points durs est possible dans la mesure où des données relatives à différents environnements (milieu urbain, péri-urbain et milieu ouvert) sont traitées. Dans cet optique des essais ont été réalisés dans la région parisienne, à Beynes. Cette zone a permis de mettre en évidence le fonctionnement du RS-SLAM-FMT (en post-traitement) dans des conditions et environnements variés.

Le tableau B.1 résume la liste des différents essais réalisés. Deux de ces essais (A.1.4 et A.3.4) sont présentés dans les sections 3.2 et 3.3 du chapitre 5. Cette annexe va donc fournir le descriptif des autres essais.

### 2 Essai A.1.3 : cas simple avec rebouclage court

La figure B.1 montre une vue aérienne de la zone couverte et du tracé.

Les caractéristiques de l'essai sont résumées dans le tableau B.2.

La figure B.2 présente un comparatif visuel entre les SLAM RS-SLAM-FMT et SLAM SICK.

En conclusion, cet essai montre que l'algorithme RS-SLAM-FMT fournit des résultats en concordance avec ceux d'autres algorithmes de SLAM et d'autres capteurs extéroceptifs (ici un LIDAR). De plus, les pièges naturels de la zone tels que les saturations du capteur liées à des bâtiments en tôle et les forts dénivelés n'ont pas réussi à mettre en échec la méthode. Il faut néanmoins rappeler que tout ceci a été effectué à une vitesse très faible (de l'ordre de 5 km/h) et sans objets mobiles présents dans la scène.

### 3 Essai A.2.2 : environnement très dense

La figure B.3 montre une vue aérienne de la zone couverte et le parcours effectué par le véhicule.

Les caractéristiques de l'essai sont résumées dans le tableau B.3.

TABLE B.1 – Liste des essais effectués en environnement naturel.

Ref.	Intitulé	Densité de l'environnement	Type d'environnement	Changement attitude
<b>A.1.</b>	<b>Mesure performance cas simple</b>			
A.1.1	Cas simple sans rebouclage, court	Environnement avec différents objets, non dense et non vide	Ouvert, Chemin, etc.	Sol plan favorable
A.1.2	Cas simple sans rebouclage, long	Environnement avec différents objets, non dense et non vide	Ouvert, Chemin, etc.	Sol plan favorable
A.1.3	Cas simple avec rebouclage, court	Environnement avec différents objets, non dense et non vide	Ouvert, Chemin, etc.	Sol plan favorable
A.1.4	Cas simple avec rebouclage, long	Environnement avec différents objets, non dense et non vide	Ouvert, Chemin, etc.	Sol plan favorable
<b>A.2.</b>	<b>Limitations</b>			
A.2.1	Environnement très peu chargé	Densité très faible d'obstacles (champ)	Ouvert	Sol plan
A.2.2	Environnement très dense	Densité forte, parcours en chemin forestier	Forêt	Sol plan
A.2.3	Environnement très dense avec bâtiments	Densité forte	Ouvert	Sol plan
A.2.4	Environnement glissant	Moyen	Ouvert	Sol plan
<b>A.3.</b>	<b>Influence des mouvements du robot</b>			
A.3.1	Changement d'attitude	Moyen	Ouvert	Sol avec pentes
A.3.2	Mouvement très favorable	Moyen	Ouvert, chemin, etc.	Sol plan
A.3.3	Mouvement classique	Moyen	Ouvert, chemin, etc.	Sol plan
A.3.4	Mouvement rapide et brusque	Moyen	Ouvert, chemin, etc.	Sol plan

La figure B.4 présente un comparatif visuel entre les SLAM RS-SLAM-FMT et la vérité terrain. Le SLAM-SICK n'ayant pas fonctionné dans ce test, il ne peut être comparé.

En conclusion, cet essai montre que l'algorithme RS-SLAM-FMT fonctionne dans cet environnement. Les pièges naturels de la zone tels que les zones ambiguës de forte densité (forêt) n'ont pas réussi à mettre en échec la méthode. Il faut néanmoins rappeler que tout ceci a été effectué à une vitesse très faible (de l'ordre de 5 km/h) et sans objets mobiles présents dans la scène.

## 4 Essai A.3.1 : changement d'attitude

La figure B.5 montre une vue aérienne de la zone couverte, on retrouve en rouge le parcours du véhicule.



FIGURE B.1 – Vue aérienne du parcours. Le parcours du véhicule est indiqué en rouge (vérité terrain GPS-RTK).

TABLE B.2 – Caractéristiques de l'essai A.1.3.

caractéristiques radar	
Nombre de panoramiques	102
Temps de cycle	1 s
Nombre de panoramiques sommés pour réduction du speckle	3
Déconvolution	OUI
caractéristiques du parcours	
Distance	350 m
Bouclage	OUI
Pente	OUI
Zones denses	zone de saturation
Zones de faible densité	NON
caractéristiques RS-SLAM-FMT	
Distance minimum entre panoramiques	6 m
Vitesse moyenne	5 km/h
Gestion des boucles	OUI

Les caractéristiques de l'essai sont résumées dans le tableau B.4.

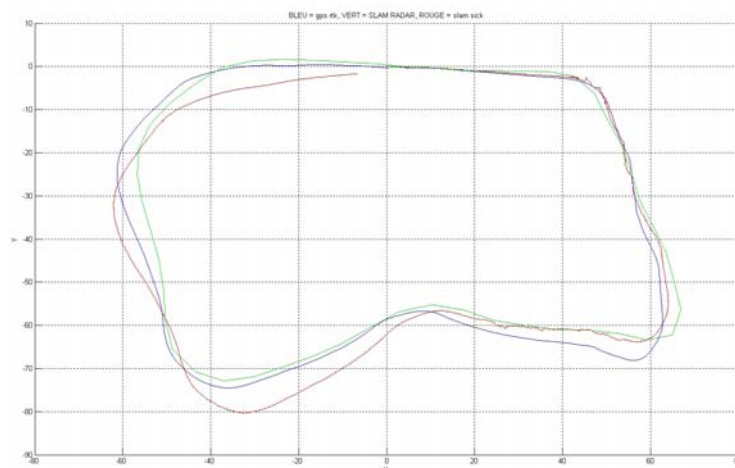


FIGURE B.2 – En rouge la vérité terrain, en bleu le RS-SLAM-FMT, en vert le SLAM SICK.



FIGURE B.3 – Vue aérienne du parcours. Le parcours du véhicule est indiqué en rouge (vérité terrain GPS-RTK).

La figure B.6 présente un comparatif visuel entre les SLAM RS-SLAM-FMT et la vérité terrain. Le SLAM-SICK n'a pas fonctionné dans ces conditions.

En conclusion, cet essai montre que l'algorithme RS-SLAM-FMT fonctionne. Les déplacements en pente avec glissement du robot n'ont pas réussi à mettre en échec le procédé. Il faut néanmoins rappeler que tout ceci a été effectué à une vitesse très faible (de l'ordre de 5 km/h) et sans objets mobiles présents dans la scène.

## 5 Essai A.3.2 : influence des mouvements du robot

La figure B.7 montre une vue aérienne de la zone, avec le parcours couvert.

Les caractéristiques de l'essai sont résumées dans le tableau B.5.

TABLE B.3 – Caractéristiques de l’essai A.2.2.

caractéristiques radar	
Nombre de panoramiques	78
Temps de cycle	1 s
Nombre de panoramiques sommés pour réduction du speckle	3
Déconvolution	OUI
caractéristiques du parcours	
Distance	713 m
Bouclage	OUI
Pente	NON
Zones denses	Très dense (forêt)
Zones de faible densité	NON
caractéristiques RS-SLAM-FMT	
Distance minimum entre panoramiques	6 m
Vitesse moyenne	5 km/h
Gestion des boucles	OUI

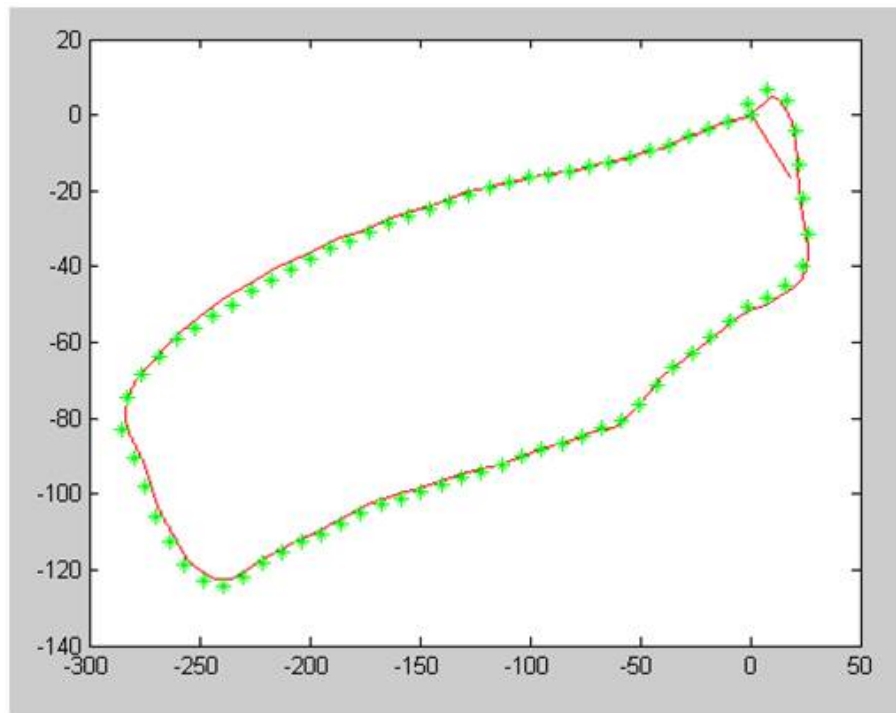


FIGURE B.4 – En rouge la vérité terrain (GPS RTK), en vert le RS-SLAM-FMT.

La figure B.8 présente un comparatif visuel entre les SLAM RS-SLAM-FMT et SLAM SICK.

En conclusion, cet essai montre que l’algorithme RS-SLAM-FMT donne des résultats en concordance avec ceux fournis par d’autres algorithmes de SLAM et d’autres capteurs extéroceptifs (ici un LIDAR). De plus, les mouvements variés du véhicule n’ont pas réussi à mettre en échec la méthode. Il faut néanmoins rappeler que tout ceci a été effectué à une vitesse très faible (de l’ordre de 5 km/h) et sans objets mobiles présents dans la scène.



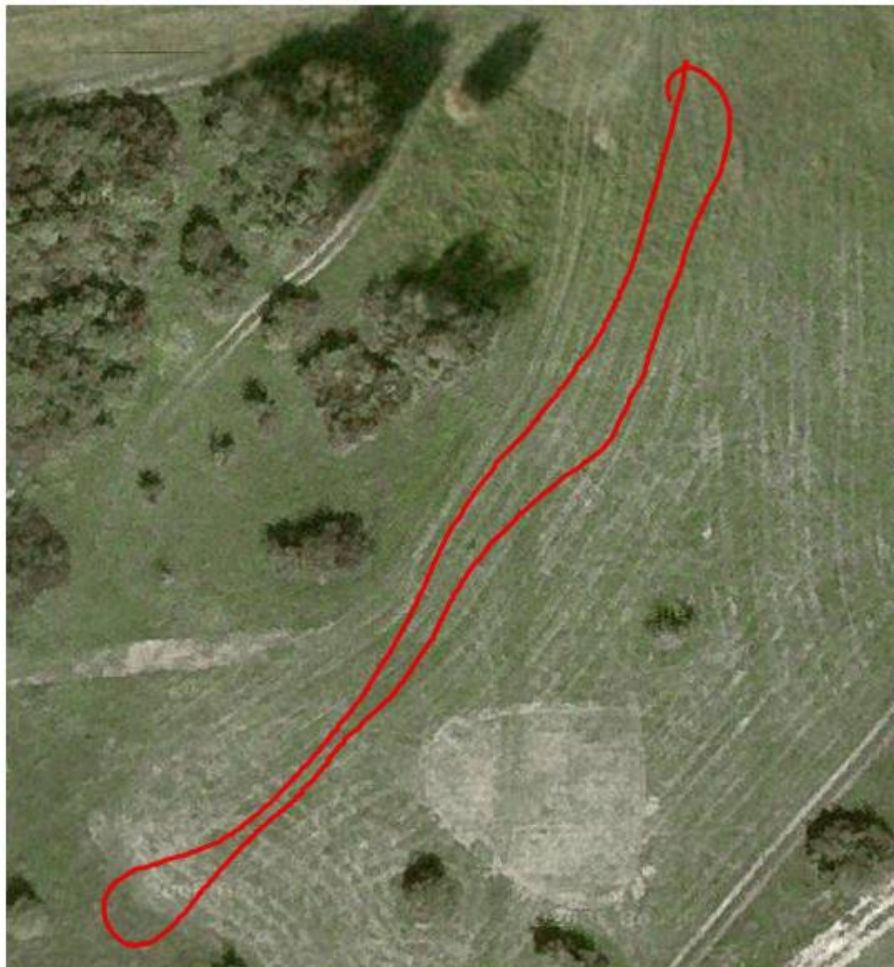


FIGURE B.5 – Vue aérienne du parcours. Le parcours du véhicule est indiqué en rouge (vérité terrain GPS-RTK).

TABLE B.4 – Caractéristiques de l'essai A.3.1.

caractéristiques radar	
Nombre de panoramiques	49
Temps de cycle	1 s
Nombre de panoramiques sommés pour réduction du speckle	3
Déconvolution	OUI
caractéristiques du parcours	
Distance	270 m
Bouclage	OUI
Pente	OUI, variée et avec glissement du robot
Zones denses	NON
Zones de faible densité	NON
caractéristiques RS-SLAM-FMT	
Distance minimum entre panoramiques	6 m
Vitesse moyenne	5 km/h
Gestion des boucles	OUI

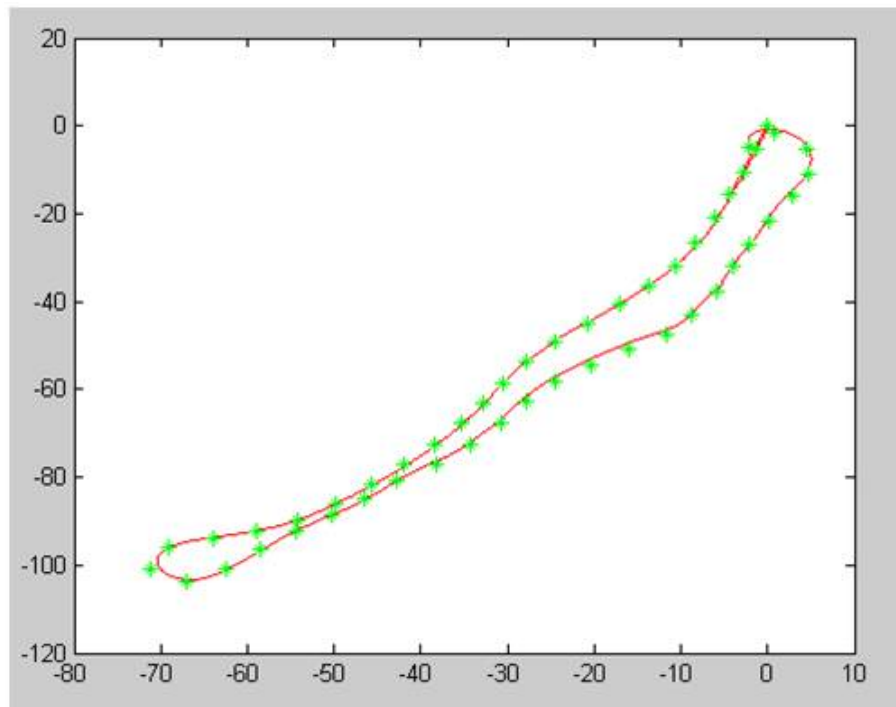


FIGURE B.6 – En rouge la vérité terrain (GPS RTK), en vert le RS-SLAM-FMT.

TABLE B.5 – Caractéristiques de l'essai A.3.2.

caractéristiques radar	
Nombre de panoramiques	96
Temps de cycle	1 s
Nombre de panoramiques sommés pour réduction du speckle	3
Déconvolution	OUI
caractéristiques du parcours	
Distance	326 m
Bouclage	OUI
Pente	NON
Zones denses	NON
Zones de faible densité	NON
caractéristiques RS-SLAM-FMT	
Distance minimum entre panoramiques	6 m
Vitesse moyenne	5 km/h
Gestion des boucles	OUI

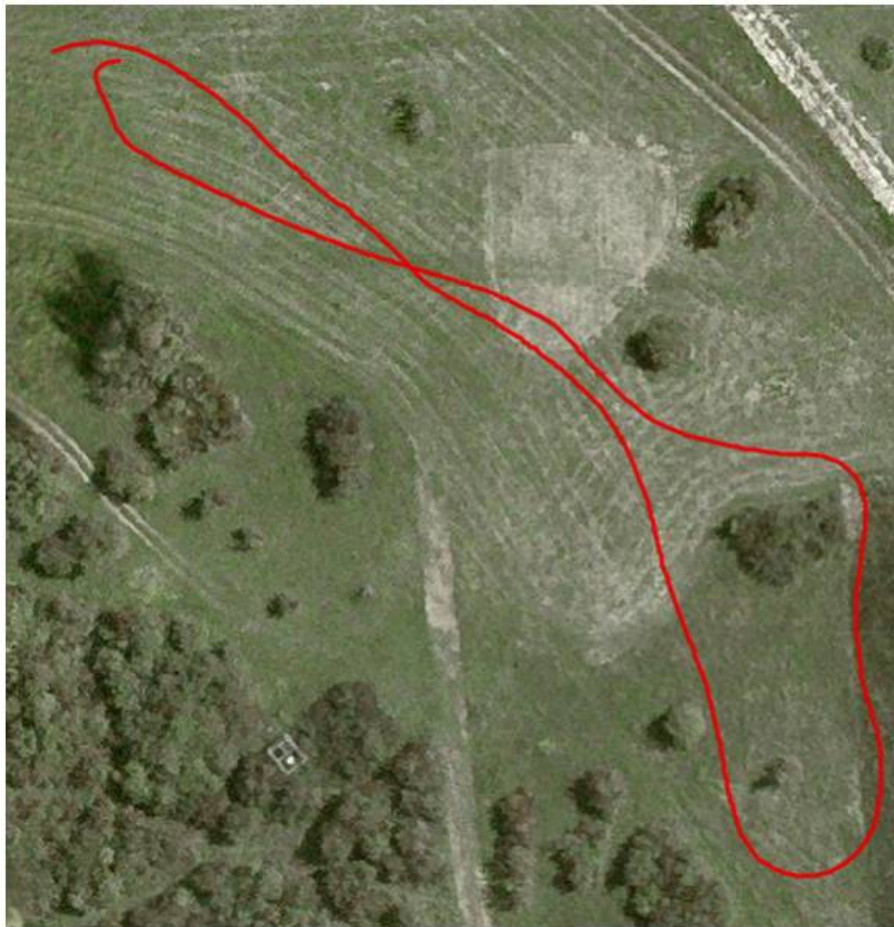


FIGURE B.7 – Vue aérienne du parcours. Le parcours du véhicule est indiqué en rouge (vérité terrain GPS-RTK).



FIGURE B.8 – En rouge la vérité terrain, en bleu le RS-SLAM-FMT, en vert le SLAM SICK.

# Annexe C

## Portage temps réel du RS-SLAM-FMT

### 1 Introduction

Le R-Trooper est un véhicule d'exploration de la société THALES. Il devra à terme accueillir l'application SLAM temps réel.

Il utilise une plateforme multiprocesseur nommée RTMaps. Progiciel de la société INTEMPORA pour le développement et la mise en œuvre d'applications temps réel multi-capteurs.

Ce logiciel supporte un grand nombre de capteurs et d'actionneurs et il dispose d'une architecture générale (cf. figure C.1) qui utilise un puissant moteur de développement écrit en C++. Afin de garantir un fonctionnement temps réel optimisé, il autorise la distribution des tâches à plusieurs processeurs. Un maître gère l'application globale, une horloge unique supervise et synchronise les esclaves (cf. figure C.2). Enfin, il offre la possibilité d'intégrer facilement des fonctionnalités incorporées dans des DLL compatibles Visual C++.

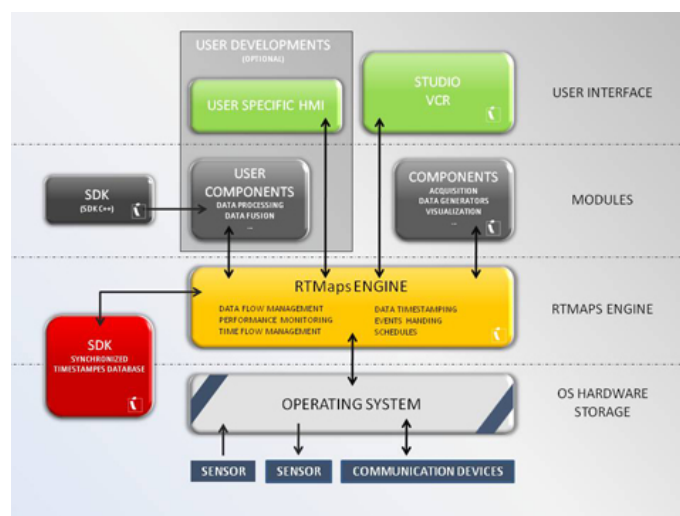


FIGURE C.1 – Architecture générale RTMaps.

Notre projet consiste à créer, à partir des codes RS-SLAM-FMT qui ont été

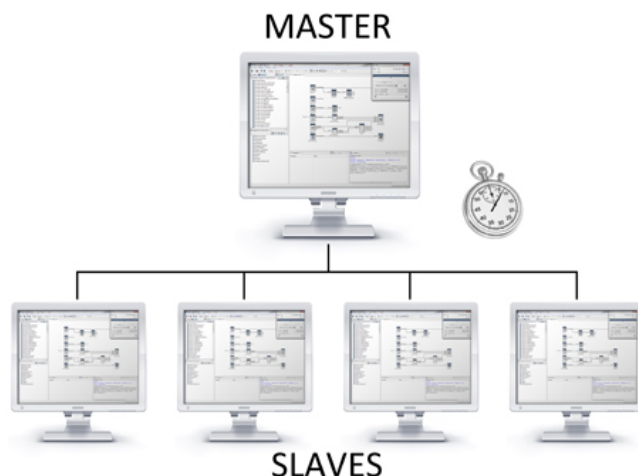


FIGURE C.2 – Distribution des tâches sous RTMaps.

validés sous Matlab, une DLL compatible Visual C++. Cette dernière devra en premier lieu fonctionner sur des données rejouées sur un PC traditionnel, puis sur la plateforme RTMaps. L'objectif est d'offrir des temps de calculs inférieurs à la seconde afin de pouvoir utiliser le radar K2pi en fonctionnement temps réel (ce dernier fournit un panoramique radar toutes les secondes).

## 2 Architecture logicielle développée

Pour comprendre l'architecture mise en œuvre, il faut en premier lieu bien comprendre la façon dont les signaux asynchrones sont traités. La figure C.3 montre une représentation graphique des signaux et la façon dont ils vont être utilisés.

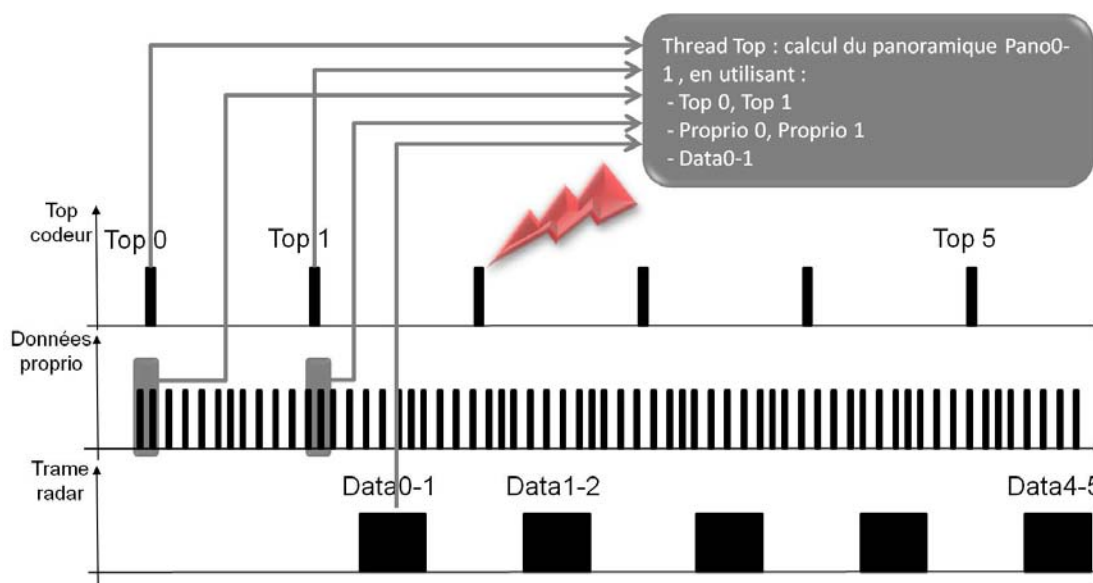


FIGURE C.3 – Représentation temporelle des signaux radars et proprioceptifs.



**Top codeur :** le radar possède un codeur angulaire qui indique la position angulaire relative de chaque faisceau. Le « Top codeur » correspond à l'instant de démarrage des acquisitions. Il va fournir un référentiel de temps sur les débuts d'acquisition de chaque panoramique (correspond à un tour complet de l'antenne radar, soit 360 faisceaux).

**Données proprioceptives :** indique le fait que les informations odométrique et gyrométrique arrivent toutes les 20 (Gyromètre) à 100 ms (odomètre).

**Trame radar :** **Data0-1** correspond à la sortie sur le réseau Ethernet des 360 faisceaux radar.

Comme l'indique la figure C.3, si l'on souhaite construire le panoramique filtré **PanoFiltré0-1**, il faut déclencher un processus à l'instant **Top 2** qui va utiliser :

- les données proprioceptives correspondants aux instant **Top 0** et **Top 1** ;
- les données radar **Data0-1**.

A partir de l'étude temporelle des signaux une architecture logicielle a été mise en place afin de respecter ce fonctionnement (cf. figure C.4).

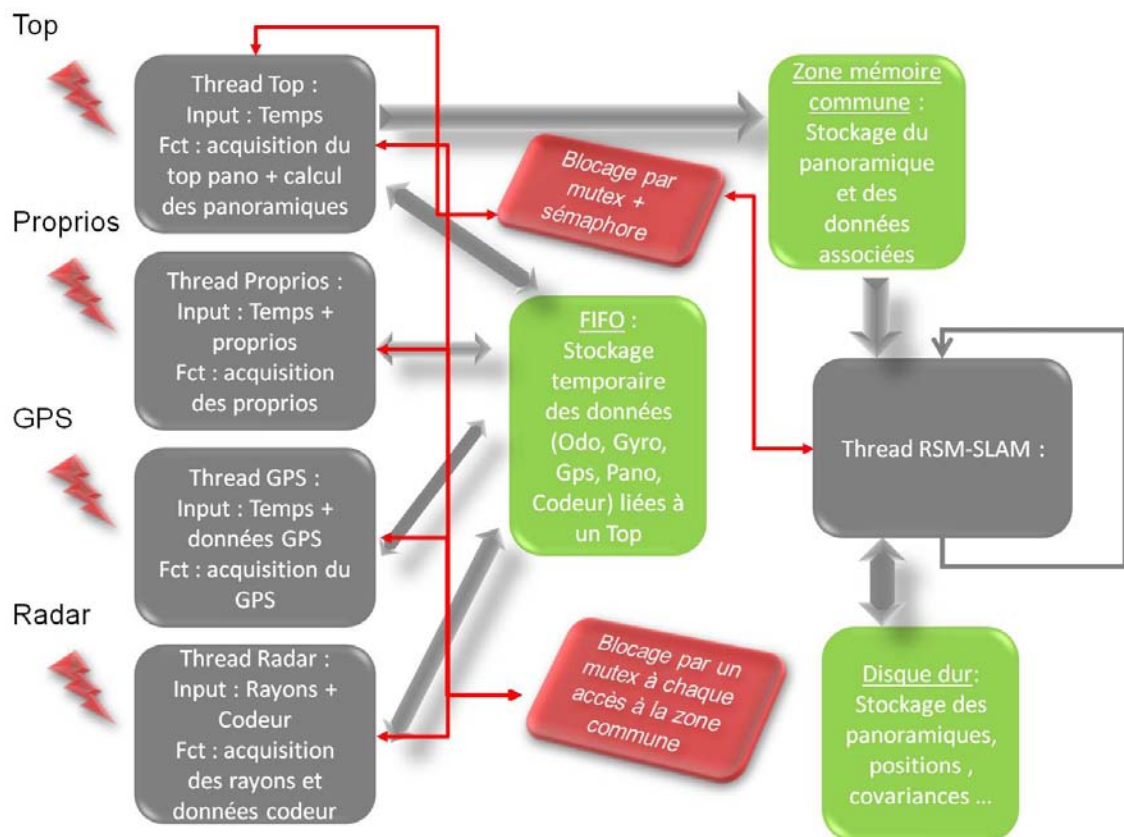


FIGURE C.4 – Architecture logicielle mise en place pour respecter le flux asynchrone des données (proprioceptives et radar).

La figure C.4 est constituée de cinq *threads*, c'est-à-dire de cinq processus indépendants qui seront déclenchés soit de façon événementielle soit de façon périodique.

Le lien entre ces processus se situe dans des zones mémoires communes. Ces zones ne pouvant être accessibles par plusieurs *threads* en même temps, un procédé de *mutex*<sup>1</sup> est mis en place. Le tableau C.1 liste les caractéristiques des ces différents processus.

TABLE C.1 – Caractéristiques des processus.

Nom du <i>thread</i>	Type de déclenchement	Entrées/Sorties	Fonctionnalité
<b>Top</b>	A chaque arrivée d'un Top	Temps du top en entrée, panoramique filtré en sortie	Calcule un panoramique filtré en moins d'une seconde
<b>Proprios</b>	A chaque arrivée d'un proprio	valeurs et instant d'arrivée des proprios en entrée, pas de sortie	Enregistre les valeurs des proprios
<b>GPS</b>	A chaque arrivée d'une trame GPS	valeurs et instant d'acquisition des données GPS en entrée, pas de sortie	Enregistre les valeurs GPS comme vérité terrain
<b>Radar</b>	A chaque arrivée d'une trame faisceau radar	valeurs codeur et signaux radar en entrée, pas de sortie	Enregistre les valeurs
<b>RS-SLAM-FMT</b>	Si un panoramique filtré est présent	Panoramique filtré, valeurs des proprios, valeur GPS, en entrée. Estimation des positions et des covariances associées, en sortie	Réalise le SLAM par l'algorithme RS-SLAM-FMT

Seuls deux processus vont occuper une part importante du temps processeur. Tout d'abord le **Thread Top** qui réalise le calcul du panoramique filtré, et d'autre part le **Thread RS-SLAM-FMT** qui effectue le SLAM. Ils seront donc placés sur deux cœurs différents. De cette façon, le procédé a donc une seconde pour calculer un panoramique filtré et une seconde pour effectuer le RS-SLAM-FMT.

### 3 Conclusion

Cet annexe donne des information sur l'architecture logicielle développée. Elle fonctionne correctement sur PC, par contre son implémentation sous RTMaps n'est pas terminée.

---

1. Un mutex est une technique utilisée en informatique pour obtenir un accès exclusif à des ressources partagées.

# Bibliographie

- Shlomo Argamon-engelson. Using image signatures for place recognition. *Pattern Recognition Letters*, 19:941–951, 1998.
- A. Averbuch, R. R. Coifman, D.L. Donoho, M. Elad et M. Israeli. Accurate and fast discrete polar fourier transform. Dans *in Proc. 37th Asilomar Conf. Signals, Systems & Computers*, pages 1933–1937, 2003.
- N. Ayache et O. D. Faugeras. *Maintaining representations of the environment of a mobile robot*, pages 205–220. Springer-Verlag New York, Inc., New York, NY, USA, 1990. ISBN 0-387-97240-4. URL <http://dl.acm.org/citation.cfm?id=93002.93296>.
- Nicholas Ayache et Olivier D. Faugeras. Building, registrating, and fusing noisy visual maps. *International Journal of Robotic Research*, 7:45–65, 1988.
- T. Bailey. *Mobile Robot Localisation and Mapping in Extensive Outdoor Environments*. Thèse de doctorat, University of Sydney, Australian Centre for Field Robotics, 2002.
- T. Bailey et H.F. Durrant-Whyte. Simultaneous Localization and Mapping : Part II - State of the Art. *Robotics & Aut. Magazine*, 2006.
- Tim Bailey, Eduardo Mario Nebot, Julio Rosenblatt et Hugh F. Durrant-whyte. Data association for mobile robot navigation : A graph theoretic approach. Dans *International Conference on Robotics and Automation*, pages 2512–2517, 2000.
- Tim Bailey, Juan Nieto, José E. Guivant, Michael Stevens et Eduardo Mario Nebot. Consistency of the ekf-slam algorithm. Dans *International Conference on Intelligent Robots and Systems - IROS*, pages 3562–3568, 2006.
- Yaakov Bar-Shalom, Thiagalingam Kirubarajan et X.-Rong Li. *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc., New York, NY, USA, 2002. ISBN 0471221279.
- Kristopher R. Beevers et Wesley H. Huang. Loop closing in topological maps. Dans *International Conference on Robotics and Automation*, pages 4367–4372, 2005.
- Ola Bengtsson et Albert-Jan Baerveldt. Robot localization based on scan-matching—estimating the covariance matrix for the idc algorithm. *Robotics and Autonomous Systems*, 44(1):29 – 40, 2003. ISSN 0921-8890. URL <http://www.sciencedirect.com/science/article/pii/S0921889003000083>.



- Paul J. Besl et Neil D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:239–256, 1992.
- Michael Bosse, Paul M. Newman, John J. Leonard, Martin Soika, Wendelin Feiten et Seth J. Teller. An atlas framework for scalable mapping. Dans *International Conference on Robotics and Automation*, pages 1899–1906, 2003.
- Michael Bosse et Robert Zlot. Map matching and data association for large-scale two-dimensional laser scan-based slam. *International Journal of Robotic Research*, 27:667–691, 2008.
- E. O. Brigham. *The fast fourier transform and its applications*. Facsimile Edition, Prentice-Hall, 1988.
- Lisa Gottesfeld Brown et Essa Basaeed. A survey of image registration techniques. *ACM Computing Surveys*, 24, 1992.
- Wolfram Burgard, Dieter Fox et Sebastian Thrun. An efficient fastslam algorithm for generating maps of large-scale cyclic environments from raw laser range measurements. Dans *International Conference on Intelligent Robots and Systems*, 2003.
- Antoni Burguera, Yolanda González et Gabriel Oliver. Underwater scan matching using a mechanical scanned imaging sonar. Dans *IFAC Symposium on Intelligent Autonomous Vehicles (IAV)*, 2010.
- George Casella et Christian Robert. Rao-blackwellisation of sampling schemes. *Biometrika*, 83:81–94, 1996.
- J. A. Castellanos, J. D. Tard'os et G. Schmidt. Building a global map of the environment of a mobile robot : The importance of correlations. Dans *International Conference on Robotics and Automation*, 1997.
- Jose A. Castellanos, J. M. M. Montiel, J. Neira et J. D. Tardos. The spmap : a probabilistic framework for simultaneous localization and map building. *IEEE Transactions on Robotics and Automation*, 15:948–952, 1999.
- E. De Castro et C. Morandi. Registration of translated and rotated images using finite fourier transforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9:700–703, 1987.
- Andrea Censi. An accurate closed-form estimate of ICP's covariance. Dans *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3167–3172, Rome, Italy, avril 2007a. URL <http://purl.org/censi/2006/icpcov>.
- Andrea Censi. On achievable accuracy for range-finder localization. Dans *International Conference on Robotics and Automation*, pages 4170–4175, 2007b.
- Andrea Censi. An icp variant using a point-to-line metric. Dans *International Conference on Robotics and Automation*, pages 19–25, 2008.

- Andrea Censi, Luca Iocchi et Giorgio Grisetti. Scan matching in the hough domain. Dans *International Conference on Robotics and Automation*, pages 2739–2744, 2005.
- R. Chatila et J. P. Laumond. Position referencing and consistent world modeling for mobile robots. Dans *Proceedings of the 2nd IEEE International Conference on Robotics and Automation (ICRA)*, pages 138–145, St. Louis, March 1985.
- Paul Checchin, Franck Gérossier, Christophe Blanc, Roland Chapuis et Laurent Trassoudaine. Radar scan matching slam using the fourier-mellin transform. Dans Andrew Howard, Karl Iagnemma et Alonzo Kelly, éditeurs, *Field and Service Robotics*, volume 62 de *Springer Tracts in Advanced Robotics*, pages 151–161. Springer Berlin / Heidelberg, 2010. ISBN 978-3-642-13407-4. URL [http://dx.doi.org/10.1007/978-3-642-13408-1\\_14](http://dx.doi.org/10.1007/978-3-642-13408-1_14). 10.1007/978-3-642-13408-1\_14.
- S. Clark. *Autonomous Land Vehicle Navigation Using Millimetre Wave Radar*. Thèse de doctorat, University of Sydney, Australian Centre for Field Robotics, 1999.
- Steve Clark et Hugh F. Durrant-whyte. Autonomous land vehicle navigation using millimeter wave radar. Dans *International Conference on Robotics and Automation*, pages 3697–3702, 1998.
- David M. Cole et Paul M. Newman. Using laser range data for 3d slam in outdoor environments. Dans *International Conference on Robotics and Automation*, pages 1556–1563, 2006.
- I. J. Cox. Blanche-an experiment in guidance and navigation of an autonomous robot vehicle. *IEEE Transactions on Robotics and Automation*, 7:193–204, 1991.
- Ingemar J. Cox et John J. Leonard. Modeling a dynamic environment using a bayesian multiple hypothesis approach. *Artificial Intelligence*, 66:311–344, 1994.
- M. Csorba, J. K. Uhlmann et H. F. Durrant-Whyte. New approach to simultaneous localization and dynamic map building. Dans S. A. Speigle, éditeur, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 2738 de *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 26–36, mai 1996.
- Frank Dellaert et Michael Kaess. Square root sam : Simultaneous localization and mapping via square root information smoothing. *International Journal of Robotic Research*, 25:1181–1203, 2006.
- Gamini Dissanayake, Paul M. Newman, Steven Clark, Hugh F. Durrant-Whyte et M. Csorba. A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions on Robotics and Automation*, 17:229–241, 2001.
- H. F. Durrant-Whyte. Uncertain geometry in robotics. *IEEE Journal on Robotics and Automation*, 4:23–31, 1988.
- H.F. Durrant-Whyte et T. Bailey. Simultaneous Localization and Mapping : Part I - The Essential Algorithms. *Robotics & Automation Magazine*, 9, 2006.

- A. Elfes. A sonar-based mapping and navigation system. Dans *International Conference on Robotics and Automation*, 1986.
- Alberto Elfes. Using occupancy grids for mobile robot perception and navigation. *IEEE Computer*, 22:46–57, 1989.
- Carlos Estrada, José Neira et Juan D. Tardós. Hierarchical slam : Real-time accurate mapping of large environments. *IEEE Transactions on Robotics*, 21:588–596, 2005.
- Ryan M. Eustice, Hanumant Singh et John J. Leonard. Exactly sparse delayed-state filters for view-based slam. *IEEE Transactions on Robotics*, 22:1100–1114, 2006.
- A. J. Fitch, Alexander Kadyrov, William J. Christmas et Josef Kittler. Orientation correlation. Dans *British Machine Vision Conference*, 2002.
- Andrew W. Fitzgibbon. Robust registration of 2d and 3d point sets. Dans *British Machine Vision Conference*, 2001.
- Hassan Foroosh, Josiane Zerubia et Marc Berthod. Extension of phase correlation to subpixel registration. *IEEE Transactions on Image Processing*, 11:188–200, 2002.
- U. Frese et G. Hirzinger. Simultaneous localization and mapping - a discussion. Dans *Reasoning with Uncertainty in Robotics*, 2001.
- U. Frese, P. Larsson et T. Duckett. A Multilevel Relaxation Algorithm for Simultaneous Localization and Mapping. *IEEE Transactions on Robotics and Automation*, 21(2), avril 2005.
- Udo Frese. Treemap : An  $o(\log n)$  algorithm for indoor simultaneous localization and mapping. *Autonomous Robots*, 21:103–122, 2006.
- Udo Frese. Efficient 6-dof slam with treemap as a generic backend. *Proceedings 2007 IEEE International Conference on Robotics and Automation*, (April):4814–4819, 2007. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4209839>.
- Rafael C. Gonzalez et Richard E. Woods. *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd édition, 2001. ISBN 0201180758.
- Giorgio Grisetti, Cyrill Stachniss et Wolfram Burgard. Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling. Dans *International Conference on Robotics and Automation*, pages 2432–2437, 2005.
- Giorgio Grisetti, Cyrill Stachniss et Wolfram Burgard. Non-linear constraint network optimization for efficient map learning. *IEEE Transactions on Intelligent Transportation Systems*, 2009.
- Franck Gérossier, Paul Checchin, Christophe Blanc, Roland Chapuis et Laurent Trassoudaine. Trajectory-oriented ekf-slam using the fourier-mellin transform applied to microwave radar images. Dans *International Conference on Intelligent Robots and Systems - IROS*, pages 4925–4930, 2009.

- J. Guivant. *Efficient Simultaneous Localization and Mapping in Large Environments*. Thèse de doctorat, University of Sydney, Australian Centre for Field Robotics, 2002.
- José E. Guivant et Eduardo Mario Nebot. Optimization of the simultaneous localization and map-building algorithm for real-time implementation. *IEEE Transactions on Robotics and Automation*, 17:242–257, 2001.
- Xiaoxin Guo, Zhiwen Xu, Yinan Lu, Zhanhui Liu et Yunjie Pang. *Image Registration Based on Pseudo-Polar FFT and Analytical Fourier-Mellin Transform*. 2005.
- J. Gutmann et Kurt Konolige. Incremental mapping of large cyclic environments. Dans *Proc. IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, pages 318–325, Monterey, California, 1999.
- Shoudong Huang et Gamini Dissanayake. Convergence analysis for extended kalman filter based slam. Dans *International Conference on Robotics and Automation*, pages 412–417, 2006.
- R. Jaulmes, Eric Moliné et Jonathan Obriet Leclef. Towards a quantitative evaluation of simultaneous. Dans *Proceedings of the 9th Workshop on Performance Metrics for Intelligent Systems*, PerMIS '09, pages 57–64, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-747-9. URL <http://doi.acm.org/10.1145/1865909.1865922>.
- Patric Jensfelt et Steen Kristensen. Active global localization for a mobile robot using multiple hypothesis tracking. *IEEE Transactions on Robotics and Automation*, 17:748–760, 2001.
- Ebi Jose et Martin David Adams. Millimetre wave radar spectra simulation and interpretation for outdoor slam. Dans *International Conference on Robotics and Automation*, pages 1321–1326, 2004.
- Michael Kaess, Ananth Ranganathan et Frank Dellaert. isam : Fast incremental smoothing and mapping with efficient data association. Dans *International Conference on Robotics and Automation*, pages 1670–1677, 2007.
- Y. Keller, A. Averbuch et M. Israeli. A pseudopolar fft technique for translation, rotation and scale-invariant image registration. *IEEE Transactions on Image Processing*, 14:12–22, 2005.
- Kurt Konolige. Large-scale map-making. Dans *National Conference on Artificial Intelligence*, pages 457–463, 2004.
- Kurt Konolige et Ken Chou. Markov localization using correlation. Dans *International Joint Conference on Artificial Intelligence*, pages 1154–1159, 1999.
- Kurt Konolige, Giorgio Grisetti, Rainer Kümmerle, Wolfram Burgard, Benson Limketkai et Régis Vincent. Sparse pose adjustment for 2d mapping. Dans *IROS*, Taipei, Taiwan, 10/2010 2010.
- C. D. Kuglin et D. C. Hines. The phase correlation image alignment method. Dans *Proceedings of IEEE Conf. Cybernetics and Soc.*, pages 163–165, september 1975.

- Benjamin Kuipers, Joseph Modayil, Patrick Beeson, Matt Macmahon et Francesco Savelli. Local metrical and global topological maps in the hybrid spatial semantic hierarchy. Dans *International Conference on Robotics and Automation*, pages 4845–4851, 2004.
- Benjamin Kuipers et Yung tai Byun. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and Autonomous Systems*, 8:47–63, 1991.
- W. Lawton. A new polar fourier transform for computer-aided tomography and spotlight synthetic aperture radar. *"IEEE Transactions on Acoustics, Speech, and Signal Processing"*, 36:931–933, 1988.
- J. J. Leonard et H. F. Durrant-whyte. Simultaneous map building and localization for an autonomous mobile robot. Dans *International Conference on Intelligent Robots and Systems*, 1991a.
- J. J. Leonard et H. F. Durrant-whyte. Simultaneous map building and localization for an autonomous mobile robot. Dans *International Conference on Intelligent Robots and Systems*, 1991b.
- Tod S. Levitt et Daryl T. Lawton. Qualitative navigation for mobile robots. *Artificial Intelligence*, 44:305–360, 1990.
- Feng Lu. *Shape registration using optimization for mobile robot navigation*. Thèse de doctorat, University of Toronto, Toronto, Ont., Canada, Canada, 1995. UMI Order No. GAXNN-07342.
- Feng Lu et Evangelos Milios. Robot pose estimation in unknown environments by matching 2d range scans. *Journal of Intelligent and Robotic Systems*, 18:249–275, 1994.
- B. Marcel, M. Briot et R. Murrieta (R.). Calcul de translation et rotation par la transformation de fourier. *Traitement du signal*, 14(2):135–149, 1997.
- M.O. Monod. *Frequency modulated radar : a new sensor for natural environment and mobile robotics*. Thèse de doctorat, Paris VI University, Paris, France, 1995.
- Michael Montemerlo et Sebastian Thrun. Large-scale robotic 3-d mapping of urban structures. Dans *International Symposium on Experimental Robotics*, pages 141–150, 2004.
- Michael Montemerlo, Sebastian Thrun, Daphne Koller et Ben Wegbreit. Fastslam : A factored solution to the simultaneous localization and mapping problem. Dans *National Conference on Artificial Intelligence*, pages 593–598, 2002.
- P Moutarlier et R Chatila. Stochastic multisensor data fusion for mobile robot localization and environment modeling. *International Symposium on Robotics Research*, pages 85–94, 1989.
- John Mullane, Martin David Adams et Wijerupage Sardha Wijesoma. Robotic mapping using measurement likelihood filtering. *International Journal of Robotic Research*, 28:172–190, 2009.

- Andreas Nüchter, Kai Lingemann, Joachim Hertzberg et Hartmut Surmann. Heuristic-based laser scan matching for outdoor 6d slam. Dans *In Advances in artificial intelligence. 28th annual German Conf. on AI*, pages 304–319, 2005.
- José Neira et Juan D. Tardós. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*, 17:890–897, 2001.
- Paul M. Newman, David M. Cole et Kin Leong Ho. Outdoor slam using visual appearance and laser ranging. Dans *International Conference on Robotics and Automation*, pages 1180–1187, 2006.
- Paul M. Newman et Kin Leong Ho. Slam-loop closing with visually salient features. Dans *International Conference on Robotics and Automation*, pages 635–642, 2005.
- Juan Nieto, Tim Bailey et Eduardo Mario Nebot. Recursive scan-matching slam. *Robotics and Autonomous Systems*, 55:39–49, 2007.
- Juan Nieto, Jose Guivant et Eduardo Nebot. Denseslam : Simultaneous localization and dense mapping. *Int. J. Rob. Res.*, 25:711–744, August 2006. ISSN 0278-3649. URL <http://dl.acm.org/citation.cfm?id=1274648.1274651>.
- Y Noyman. Ground surface sensing through plant foliage using an fm-cw radar. *Computers and Electronics in Agriculture*, 15:181–193, 1996.
- Edwin Olson, John J. Leonard et Seth J. Teller. Fast iterative alignment of pose graphs with poor initial estimates. Dans *International Conference on Robotics and Automation*, pages 2262–2269, 2006.
- Edwin B. Olson. Real-time correlative scan matching. Dans *International Conference on Robotics and Automation*, pages 4387–4393, 2009.
- Max Pfingsthorn, Andreas Birk, Sören Schwertfeger, Heikow Bülow et Kaustubh Pathak. Maximum likelihood mapping with spectral image registration. Dans *International Conference on Robotics and Automation*, pages 4282–4287, 2010.
- B. Srinivasa Reddy et Biswanath N. Chatterji. An fft-based technique for translation, rotation, and scale-invariant image registration. *IEEE Transactions on Image Processing*, 5:1266–1271, 1996.
- Thomas Röfer. Using histogram correlation to create consistent laser scan maps. Dans *International Conference on Intelligent Robots and Systems*, 2002.
- David Ribas, Pere Ridao, Juan D. Tardós et José Neira. Underwater slam in a marina environment. Dans *International Conference on Intelligent Robots and Systems - IROS*, pages 1455–1460, 2007.
- R. Rouveure, P. Faure, P. Checchin et M.O. Monod. Mobile Robot Localization and Mapping in Extensive Outdoor Environment based on Radar Sensor - First Results. Dans *Physics in Signal and Image Processing*, Feb 2007.

- R. Rouveure, P. Faure et M.O. Monod. A new radar sensor for mobile robot localization and mapping in extensive outdoor environment. first results. Dans *International Society for Photogrammetry and Remote Sensing (ISPRS)*, page 5, june 2006.
- J. Ryde et H. Hu. Mobile robot 3D perception and mapping with multi-resolution occupancy lists. Dans *Proceedings of IEEE International Conference on Mechatronics and Automation (ICMA)*, Harbin, Heilongjiang, China, august 2007.
- Francesco Savelli et Benjamin Kuipers. Loop-closing and planarity in topological map-building. Dans *International Conference on Intelligent Robots and Systems*, 2004.
- Alan C. Schultz et William Adams. Continuous localization using evidence grids. Dans *International Conference on Robotics and Automation*, pages 2833–2839, 1998.
- Sören Schwertfeger, Adam Jacoff, Chris Scrapper, Johannes Pellenz et Alexander Kleiner. Evaluation of maps using fixed shapes : The fiducial map metric. Dans *Proc. of the Int. Workshop on Performance Metrics for Intelligent Systems (PerMIS)*, pages 344–351. NIST, 2010.
- C. M. Smith. *Integrating mapping and navigation*. Thèse de doctorat, Department of Ocean Engineering, Massachusetts Institute of Technology, Cambridge, 1998.
- R. Smith, M. Self et P. Cheeseman. *Estimating uncertain spatial relationships in robotics*, pages 167–193. Springer-Verlag New York, Inc., New York, NY, USA, 1990. ISBN 0-387-97240-4. URL <http://dl.acm.org/citation.cfm?id=93002.93291>.
- R. C. Smith et P. Cheeseman. On the representation and estimation of spatial uncertainty. *International Journal of Robotic Research*, 5:56–68, 1986.
- Randall Smith, Matthew Self et Peter Cheeseman. A stochastic map for uncertain spatial relationships. Dans *Proceedings of the 4th international symposium on Robotics Research*, pages 467–474, Cambridge, MA, USA, 1988. MIT Press. ISBN 0-262-02272-9. URL <http://dl.acm.org/citation.cfm?id=57425.57472>.
- Petr Stepan, Miroslav Kulich et Libor Preucil. Robust data fusion with occupancy grid. *IEEE Transactions on Systems, Man, and Cybernetics*, 35:106–115, 2005.
- Harold S. Stone, Michael T. Orchard, Ee-Chien Chang et Stephen A. Martucci. A fast direct fourier-based algorithm for subpixel registration of images. *IEEE Transactions on Geoscience and Remote Sensing*, 39:2235–2243, 2001.
- Juan D. Tardós, José Neira, Paul M. Newman et John J. Leonard. Robust mapping and localization in indoor environments using sonar data. *International Journal of Robotic Research*, 21:311–330, 2002.
- Sebastian Thrun. Probabilistic algorithms in robotics. *Ai Magazine*, 21:93–109, 2000.
- Sebastian Thrun et Arno Bücken. Integrating grid-based and topological maps for mobile robot navigation. Dans *National Conference on Artificial Intelligence*, pages 944–950, 1996.

- Sebastian Thrun, Wolfram Burgard et Dieter Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. *Auton. Robots*, 5:253–271, July 1998. ISSN 0929-5593. URL <http://dl.acm.org/citation.cfm?id=591469.591483>.
- Sebastian Thrun, Mark Diel et Dirk Hähnel. Scan alignment and 3d surface modeling with a helicopter platform. Dans *In Proceedings of the International Conference on Field and Service Robotics*, 2003.
- Sebastian Thrun, Yufeng Liu, Daphne Koller, Andrew Y. Ng, Zoubin Ghahramani et Hugh Durrant-Whyte. Simultaneous localization and mapping with sparse extended information filters. *The International Journal of Robotics Research*, 23(7-8):693–716, 2004.
- Sebastian Thrun et Michael Montemerlo. The graph slam algorithm with applications to large-scale mapping of urban structures. *International Journal of Robotic Research*, 25:403–429, 2006.
- Nicola Tomatis, Illah R. Nourbakhsh et Roland Siegwart. Hybrid simultaneous localization and map building : a natural integration of topological and metric. *Robotics and Autonomous Systems*, 44:3–14, 2003.
- Masahiro Tomono. A scan matching method using euclidean invariant signature for global localization and map building. Dans *International Conference on Robotics and Automation*, pages 866–871, 2004.
- Iwan Ulrich et Illah R. Nourbakhsh. Appearance-based place recognition for topological localization. Dans *International Conference on Robotics and Automation*, pages 1023–1029, 2000.
- D. Vivet, P. Checchin et R. Chapuis. Radar-only localization and mapping for ground vehicle at high speed and for riverside boat. Dans *IEEE Inter. Conf. on Robotics and Automation (ICRA)*, St Paul, Minnesota, USA, 5 2012.
- D. Vivet, F. Gérossier, P. Checchin, R. Chapuis, M.O. Monod, P. Faure, R. Rouveure et J. Morillon. SLAM application with a microwave radar sensor. *CityHome ICT-ASIA project 2008-2010*, October, 28-29 2009. Cityhome 09.
- C.-C. Wang et C. Thorpe. Simultaneous Localization And Mapping with Detection And Tracking of Moving Objects. Dans *IEEE Int. Conf. on Robotics and Automation*, May 2002.
- Christopher Wetzler et Ewald von Puttkamer. Keeping track of position and orientation of moving indoor systems by correlation of range-finder scans. Dans *International Conference on Intelligent Robots and Systems*, 1994.
- Stefan Bernard Williams. *Efficient Solutions to Autonomous Mapping and Navigation Problems*. Thèse de doctorat, University of Sydney, Australian Centre for Field Robotics, 2001.
- George Wolberg et Siavash Zokai. Robust image registration using log-polar transform. Dans *International Conference on Image Processing*, 2000.



H Xie, N Hicks, Gr Keller, H Huang et V Kreinovich. An idl/envi implementation of the fft-based algorithm for automatic image registration. *Computers & Geosciences*, 29:1045–1055, 2003.

Takehisa Yairi. Covisibility-based map learning method for mobile robots, 2004.

Brian Yamauchi. A frontier-based approach for autonomous exploration. Dans *In Proceedings of the IEEE International Symposium on Computational Intelligence, Robotics and Automation*, pages 146–151, 1997.